UNIVERSITY OF CALIFORNIA, SAN DIEGO

Algorithms For Tandem Mass Spectrometry-Based Proteomics

A dissertation submitted in partial satisfaction of the
requirements for the degree

Doctor of Philosophy

in

Computer Science

by

Ari Michael Frank

Committee in charge:

Professor Pavel A. Pevzner, Chair
Professor Steven P. Briggs, Co-Chair
Professor Vineet Bafna
Professor Sanjoy Dasgupta
Professor Glenn Tesler

2008

The dissertation of Ari Michael Frank is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2008

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Pursuing this doctorate has been a lengthy journey. I could not have done it without the support of my family. I am especially grateful to my wife, Efrat, for her patience and support during these years. I would also like to take this opportunity to apologize to our parents for being away from them for so long, and more importantly, for depriving them of the company of their granddaughter Ayellet.

I would like to express my heartfelt gratitude to my advisor Pavel Pevzner for his guidance over the years. Pavel not only provided keen insight and many useful suggestions, but also allowed me to explore new avenues, while always making sure that I did not stray off course. I would like to thank Vineet Bafana, Steve Briggs, Sanjoy Dasgupta and Glenn Tesler for serving on my committee, and for giving me helpful advice on numerous occasions.

During my tenure as a graduate student I have had the pleasure of meeting and interacting with many fine people in the bioinformatics lab. I would like to thank the following students (past and present) for helping me out on various occasions, or just for hanging around for lively discussions in the lab: Nuno Bandeira, Vikas Bansal, Ali Bashir, Natalie Castellana, Nitin Gupta, Neil Jones, Sangtae Kim, Samuel Payne, Qian Peng, Stephen Tanner, and Shaojie Zhang. I would especially like to express thanks to Stephen and Sam for the many helpful MS/MS-related discussions we had, and for their help with InsPecT; it has proven to be a valuable tool for my research. I am also grateful to Nuno for the many discussions we have had over the years on de novo sequencing and related subjects.

My research has been made possible through the hard work performed in many of our collaborating labs. I would especially like to acknowledge the Briggs lab at UCSD (special thanks to Zhouxin Shen), the Kelleher and Mizzen labs at UIUC (thank you Jim Pesavento), the Shevchenko lab at Max Planck's institute (special thanks to Patrice Waridel), Dick Smith's lab at PNNL, and the Zubarev lab at Uppsala.

I would like to acknowledge the UCSD FWGrid Project for the availability of their computational infrastructure and especially thank Ian Kaufman who helped make sure that the jobs ran smoothly. The UCSD FWGrid Project is funded in part by NSF Research Infrastructure Grant number NSF EIA-0303622.

I am especially appreciative for the financial support of the La Jolla Interfaces in Science (LJIS) Interdisciplinary Fellowship which is sponsored by the Burroughs Wellcome Fund. This work was also supported by NIH grant NIGMS 1-R01-RR16522.

Chapter 2, in full, was published as "PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling". A. Frank and P. Pevzner. *Analytical Chemistry*, 77:964-973, 2005. The dissertation author was the primary author of this paper.

Chapter 3, in full, was published as "Peptide Sequence Tags for Fast Database Search in Mass-Spectrometry". A. Frank, S. Tanner, V. Bafna, and P. Pevzner. *Journal of Proteome Research*, 4:1287-95, 2005. The dissertation author was the primary author of this paper.

Chapter 4, in full, was published as "De Novo Peptide Sequencing and Identification with Precision Mass Spectrometry". A.M. Frank, M.M Savitski, M.L. Nielsen, R.A. Zubarev, and P.A. Pevzner. *Journal of Proteome Research*, 6:114-123, 2007. The dissertation author was the primary author of this paper.

Chapter 7, in full, was published as "Clustering Millions of Mass Spectra". A.M. Frank, N. Bandeira, Z. Shen, S. Tanner, S.P. Briggs, R.D. Smith and P.A. Pevzner. *Journal of Proteome Research*. The dissertation author was the primary author of this paper.

Chapter 8, in full, was published as "Interpreting Top-Down Mass Spectra Using Spectral Alignment". A.M. Frank, J.J. Pesavento, C.A. Mizzen, N.L. Kelleher, and P.A. Pevzner. *Analytical Chemistry*, 80:2499-2505, 2008. The dissertation author was the primary author of this paper.

VITA

1993-1996	Israeli Defense Forces
1999	Bachelor of Arts, Computer Science, Technion - Israel Institute of Technology
2002	Bachelor of Arts, Biology, Technion - Israel Institute of Technology
2002	Master of Sciences, Computer Science, Technion - Israel Institute of Technology
2008	Doctor of Philosophy, Computer Science, University of California, San Diego

PUBLICATIONS

A.M. Frank, J.J. Pesavento, C.A. Mizzen, N.L. Kelleher, and P.A. Pevzner. Interpreting Top-Down Mass Spectra Using Spectral Alignment. *Analytical Chemistry*, 80:2499-2505, 2008

A.M. Frank, N. Bandeira, Z. Shen, S. Tanner, S.P. Briggs, R.D. Smith and P.A. Pevzner. Clustering Millions of Mass Spectra. *Journal of Proteome Research*, 7:113-122, 2008

N. Bandeira, D. Tsur, **A. Frank**, and P.A. Pevzner. Protein Identification by Spectral Networks Analysis. *Proc. Natl. Acad. Sci. U.S.A.*, 104:6140-6145, 2007

P. Waridel, **A. Frank**, H. Thomas, V. Surendranath, S. Sunyaev, P. Pevzner, and A. Shevchenko. Sequence similarity-driven proteomics in organisms with unknown genomes by LC-MS/MS and automated de novo sequencing. *Proteomics*, 7:2318-2329, 2007

A.M. Frank, M.M. Savitski, M.L. Nielsen, R.A. Zubarev, and P.A. Pevzner. De Novo Peptide Sequencing and Identification with Precision Mass Spectrometry. *Journal of Proteome Research*, 6:114-123, 2007

N. Wielsch, H. Thomas, V. Surendranath, P. Waridel, **A. Frank**, P. Pevzner, A. Shevchenko. Rapid validation of protein identifications with the borderline statistical confidence via de novo sequencing and MS BLAST searches. *Journal of Proteome Research*, 5:2448-2456, 2006

S. Tanner, H. Shu, **A. Frank**, L. Wang, E. Zandi, M. Mumby, P.A. Pevzner, and V. Bafna. InsPecT: Fast and Accurate Identification of Post-Translationally Modified Peptides from Tandem Mass Spectra. *Analytical Chemistry*, 77:4626-4639, 2005

A. Frank, S. Tanner, V. Bafna, and P. Pevzner. Peptide Sequence Tags for Fast Database Search in Mass-Spectrometry. *Journal of Proteome Research*, 4:1287-95, 2005

A. Frank, S. Tanner and P.A. Pevzner. Peptide Sequence Tags for Fast Database Search in Mass-Spectrometry. In *Research in Computational Molecular Biology: Proceedings of the Ninth annual International Conference on Research in Computational Molecular Bilogy (RECOMB 2005)*, 326-341, 2005

A. Frank and P. Pevzner. PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling. *Analytical Chemistry*, 77:964-973, 2005

A. Frank, D. Geiger, and Z. Yakhini. A Distance-Based Branch and Bound Feature Selection Algorithm. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, 241-248, 2003

FIELDS OF STUDY

Major Field: Computer Science

Studies in Bioinformatics.
Professor Pavel A. Pevzner

Studies in Machine Learning.
Professor Sanjoy Dasgupta
Professor Charles Elkan

ABSTRACT OF THE DISSERTATION

Algorithms For Tandem Mass Spectrometry-Based Proteomics

by

Ari Michael Frank
Doctor of Philosophy in Computer Science
University of California, San Diego, 2008
Professor Pavel A. Pevzner, Chair
Professor Steven P. Briggs, Co-Chair

Tandem mass spectrometry (MS/MS) has emerged as the leading technology for high-throughput proteomics analysis, making it possible to rapidly identify and characterize thousands of different proteins in complex biological samples. In recent years we have witnessed a dramatic increase in the capability to acquire proteomics MS/MS data. To avoid computational bottlenecks, this growth in acquisition power must be accompanied by a comparable improvement in analysis capabilities. In this dissertation we present several algorithms we developed to meet some of the major computational challenges that have arisen in MS/MS analysis. Throughout our work we continually address two (at times overlapping) problems: how to make MS/MS-based sequence identifications more accurate, and how to make the identification process work much faster.

Much of the work we present revolves around algorithms for de novo sequencing of peptides, which aims to discover the amino acid sequence of protein digests (peptides), solely from their experimental mass spectrum. We start off by describing a new scoring model which is used in our de novo sequencing algorithm called PepNovo. Our scoring scheme is based on a graphical model decomposition that describes many of the conditions that determine the intensities of fragment ions observed in mass spectra, such as

dependencies between related fragment ions and the influence of the amino acids adjacent to the cleavage site.

Besides predicting whole peptide sequences, one of the most useful applications of de novo algorithms is to generate short sequence tags for the purpose of database filtration. We demonstrate how using these tags speeds up database searches by two orders of magnitude compared to conventional methods. We extend the use of tag filtration and show that with high-resolution data, our de novo sequencing is accurate enough to enable extremely rapid identification via direct hash lookup of peptide sequences.

The vast amount of MS/MS data that has become available has made it possible to use advanced data-driven machine learning methods to devise more acute algorithms. We describe a new scoring function for peptide-spectrum matches that uses the RankBoost ranking algorithm to learn and model the influences of the many intricate processes that occur during peptide fragmentation. Our method's superior discriminatory power boosts PepNovo's performance beyond the current state-of-the-art de novo sequencing algorithms. Our score also greatly improves the performance of database search programs, significantly increasing both their speed and sensitivity. When we applied our method to the challenging task of a proteogenomic search against a six-frame translation of the human genome, we were able to significantly increase the number of peptide identifications compared to current techniques by 60%.

To help speed up MS/MS analysis, we developed a clustering algorithm that exploits the redundancy that is inherent in large mass spectrometry datasets (these often contain hundreds and even thousands of spectra of the same peptide). When applied to large MS/MS datasets on the order of ten million spectra, our clustering algorithm reduces the number of spectra by an order of magnitude, without losing peptide identifications.

Finally, we touch upon sequencing of intact proteins ("top-down" analysis), which from a computational perspective, is only in its infancy – very few algorithms have been developed for analysis of this type of data. We developed MS-TopDown, which uses the Spectral Alignment algorithm to characterize protein forms (i.e., determine the modification/mutation sites). Our algorithm can handle heavily modified proteins and can also distinguish between several isobaric protein forms present in the same spectrum.

1

The Role of Tandem Mass Spectrometry in Proteomics

1.A Introduction

In the post-genomic era, the focus has shifted towards the identification and characterization of all gene products that are expressed in a given organism [6]. This large-scale analysis of proteins, dubbed *Proteomics*, contributes greatly to the understanding of gene function, biochemistry of proteins, processes and pathways [158]. Tandem mass spectrometry (MS/MS) has emerged as the tool of choice for high-throughput identification of proteins and determination of details of their primary structures [2, 28, 231]. Recent technological breakthroughs have led to a dramatic increase in the volume of proteomics data being generated. However, analyzing this data is not a trivial task. Without developing novel, more nimble and discriminating algorithms, we can expect computational bottlenecks to restrict the scope of discoveries that can be made from experiments involving mass spectrometry. In this dissertation we present several algorithms we developed to meet some of these rising computational challenges.

1.B What is Tandem Mass Spectrometry?

A mass spectrometer is an analytical instrument widely used to measure the weight of atoms and molecules. Mass spectrometers consist of three basic parts: an *ion*

source which ionizes the molecules in the sample, a *mass analyzer* which separates the ions according to their mass-to-charge ratio (m/z), and a *detector system* which measures the abundance of ions with different mass-to-charge ratios. Though mass spectrometry has been used in analytical chemistry for over a century, only in the last two decades has the technology advanced to offer sufficiently high resolution to make mass spectrometry an effective tool for proteomics studies.

Tandem mass spectrometry (MS/MS) has become the main "workhorse" of mass spectrometry-based proteomics. Using MS/MS, one can rapidly identify and characterize a large numbers of proteins in a single run. In a method called shotgun or "bottom-up" proteomics, the proteins of interest are first digested to shorter peptides using an enzyme such as trypsin. The resulting heterogenous mixture of peptides is then separated by means of chromatography (usually 1D or 2D liquid chromatography [51]), and transferred to the mass spectrometer for analysis that takes place in two rounds. First the peptides are ionized, typically by soft ionization techniques such as electrospray ionization (ESI) [63] or matrix-assisted laser desorption/ionization (MALDI) [135]. The mass spectrometer then determines the intact peptide's mass-to-charge-ratio (m/z). In the second round, the intact peptide ions with a specific m/z are selected and fragmented to produce information about parts of the peptide, which can be used to determine the peptide's primary structure (i.e., the amino acid sequence and chemical modifications). The most commonly used fragmentation method is collision-induced dissociation (CID) [232]. In this method the analyzed molecules are accelerated and collided with inert gas molecules. During the collision, some of the kinetic energy is converted into internal energy which results in chemical bond breakage and the fragmentation of the molecule. Other fragmentation methods often used in MS/MS-based proteomics include electron capture dissociation (ECD) [247] and electron transfer dissociation (ETD) [209].

A mass spectrometer's accuracy depends on the resolving power of its mass analyzer. The most widely used analyzers are quadrupoles, ion traps, and time-of-flight (TOF) analyzers [201]. Their resolution typically gives accuracy in the range of 100 parts-per-million (ppm), which translates to an error of 0.1 Da when measuring a fragment with a mass of 1000 Da. Continuous efforts to improve mass resolution recently resulted in the breakthrough development of Fourier transform MS techniques, including magnet-based

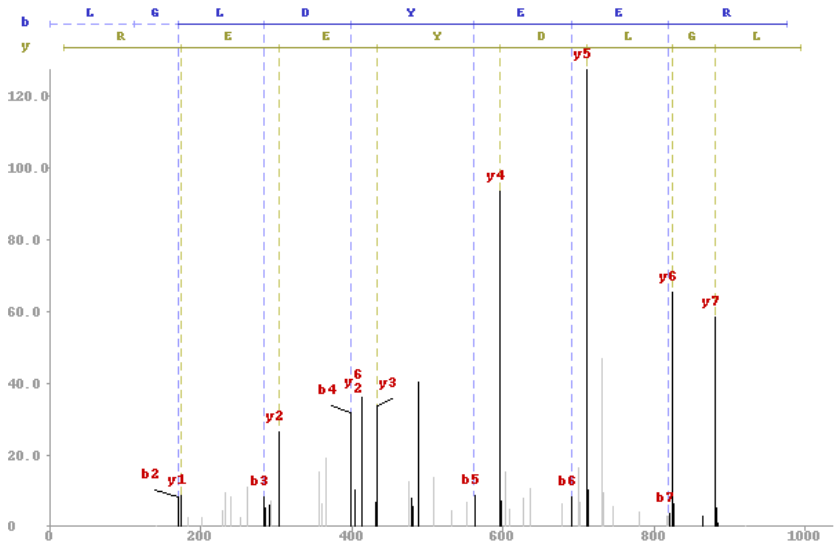


Figure 1.1: A mass spectrum of the peptide LGLDYEER. The values on the x-axis are the peak m/z values, and the y-axis holds the relative intensity. The prefix b-ions and suffix y-ions are aligned to the peptide sequence and to its reverse, respectively.

ion cyclotron resonance (ICR) instruments [140] and electrostatic FT traps (Orbitraps) [154], that improve resolution by two to three orders of magnitude as compared to conventional mass spectrometers.

A peptide's *mass spectrum*, a term which is widely used in this dissertation, consists of the m/z value of the intact peptide measured in the first MS round, which is called the *precursor mass*, and its intensity, along with the list of the fragments observed in the second round of MS, which are recorded as m/z values and their corresponding intensities (Figure 1.1 depicts a mass spectrum aligned to a peptide sequence). At times, it is possible to deduce the charge z for a peak by observing the m/z values of its natural isotopes (a series of peaks separated by values of $1/z$ indicate that the measured fragment has charge z), however often the mass spectrometer's resolution is insufficient to make such distinctions for $z > 1$. Many of the intense peaks recorded in a mass spectrum can be associated with peptide fragments, and are thus considered *signal* peaks. In addition, spectra often contain many *noise* peaks from sources other than the analyzed peptide, such as foreign chemical compounds or instrument artifacts.

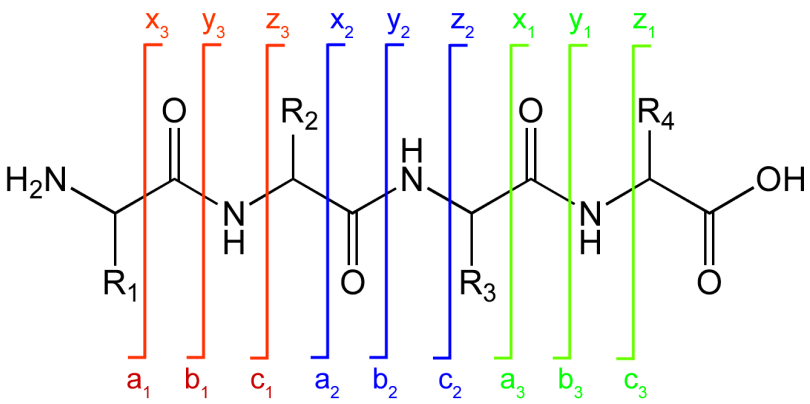


Figure 1.2: Peptide fragmentation notation.[2] N-terminal prefix fragments are designated by a_i, b_i, c_i and C-terminal suffix fragments are designated by the letters x_i, y_i, z_i. The subscript i indicates the amino that got fragmented, while the letters a, b, c and x, y, z refer to the exact location of the fragmentation relative to amino acid i's α carbon (the carbon to which the residue R_i is attached). This notation follows the conventions first described by Roepstorff and Fohlman [177] and modified by Biemann [20].

1.C Peptide Fragmentation

The process of peptide fragmentation is complex, and often involves several chemical pathways, including some that have yet to be discovered or thoroughly understood. Since mass spectra are generated by recording the products of many stochastic fragmentation events, they tend to be reproducible, i.e., repeated experiments with the same peptides produce similar looking spectra. However, due to the complexity of the fragmentation process it is often difficult to make an accurate prediction of a spectrum's "shape" (which fragment ions will be observed and with what relative intensities). Nonetheless, such knowledge can be invaluable for discovering the correct peptide sequence. Later in this dissertation we will explore how this knowledge can be learned using machine learning methods, and utilized towards improving the accuracy and sensitivity of peptide identification.

Peptide fragmentation occurs primarily along the ploy amide backbone. Only fragments that retain a charge can be detected. If the N-terminal retains a charge we observe prefix a-, b-, and c-ions, and if the C-terminal retains a charge we observe suffix x-, y-, and z-ions, as depicted in Figure 1.2. In Section 2.A.1 we examine in greater detail the types of fragment ions typically observed in the MS/MS data. Figure 1.1 depicts a mass spectrum with the dominant b- and y-ions aligned to a peptide sequence.

The relative abundance of the various ion fragments is determined by comparing the intensity of their corresponding peaks in the mass spectrum. There is much disparity in the intensities of observed fragment intensities (see Figure 1.1). Fragment intensities are influenced by many factors like the peptide's amino acid sequence, the original charge state, the method of fragmentation, etc. A lot of effort has been devoted towards understanding the processes that govern peptide fragmentation [34, 83, 153, 157, 159]. The most widely accepted model for these processes is the *mobile proton model* [48, 53, 91, 110, 206, 216, 234]. According to this model, during the ionization stage, peptides are protonated at various sites (mainly terminal amino groups and basic side chain groups), with some protanation sites being more favored than others (e.g., the amino group of arginine). Cleavage of the peptide amide bonds is initiated by migration of the sequestered charge (proton) from the initial site of protonation to an amide carbonyl oxygen along the peptide backbone. With a proton at a carbonyl oxygen, a nearby carbonyl (N-terminal to the protonated carbonyl oxygen) can serve as a nucleophile to attack the electropositive carbon of the protonated carbonyl [170, 189, 234]. In these cases the fragmentation is considered a "charge-directed" process, since it requires the involvement of a proton at the cleavage site [234].

The energy required to mobilize a proton from a basic side chain or the terminal amino group depends on the peptide's amino acid composition, with dissociation energy requirements mimicking the order of amino acid gas phase basicity [157], i.e., the energy required to mobilize a proton that is on arginine > lysine > histidine > non-basic amino acids. In some cases, such as peptides containing arginines, the energy required to mobilize the proton might be too high, which gives the possibility for alter-

[2]Image taken from `http://commons.wikimedia.org/wiki/Image:Peptide_fragmentation.gif` available under the creative commons license, see `http://creativecommons.org/licenses/by-sa/3.0/`.

native fragmentation pathways to dominate; pathways that do not involve the ionizing proton (a "charge-remote" fragmentation process). These fragmentation events often get initiated by ions containing long-chain or poly-ring structures [31]. The dissociation of protonated peptides can be described as a competition between charge-remote and charge-directed fragmentation pathways, in a complicated reaction pattern where fragment ions are formed with substantially different probabilities [157].

The activity of charge-direct and charge-remote pathways can be significantly enhanced or reduced depending on the identity of the amino acids adjacent to the peptide's cleavage site. For instance, the fragmentation of the bond N-terminal to proline is known to be very active when there is a mobile proton [23, 225]. Often this is the most active site in the peptide, and it is responsible for a very intense peak in the observed spectrum. Similar enhancement of the fragmentation, though slightly less intense, occurs if we replace the proline with a glycine residue. In contrast, when no mobile protons are available, the charge-remote pathways can dominate and cleavage is enhanced near acidic residues [103, 234].

In addition to fragmentation of the backbone, there are also pathways that lead to the peptides losing neutrally charged molecules such CO ,NH_3 or H_2O [159], which diversifies the set of observed fragment ions. Certain fragmentation pathways can also lead to rearrangement of the peptide sequence [235], which adds a significant degree of complexity to the observed mass spectra.

Understanding the peptide fragmentation processes is important for developing more sensitive algorithms for peptide identification [101, 142, 213, 234]. Several data-mining studies have been performed on large sets of mass spectra to quantify the effects various sequence-based features have on fragmentation, such as the influence the amino acids adjacent to the cleaved bond have on its observed intensity [14, 23, 101, 115, 213], the effect ceratin amino acids have on the abundance of neutral losses [142, 184, 213], and the effect of the composition of basic amino acids in the peptide [101, 115, 210]. In some cases this information was incorporated into scoring models for peptide identification [59, 191, 243]. In Chapter 5 we show novel ways in which this information can be used by data-driven machine learning methods, and create stronger scoring models that significantly improve our algorithms' performance.

1.D MS/MS-Based Peptide Identification

Until the early 1990's the primary method for identifying proteins was a chemical reagent-based method called *Edman degradation* [57]. In this method, amino acids are sequentially removed one by one from the protein's amino terminus. The sequence is determined by noting which amino acid was removed each round. This method has a slow turn-over rate, and requires the analyzed protein sample to be homogenous and in relatively large quantities. Thus this method is not well suited for large scale high-throughput proteomics studies and was quickly replaced by mass spectrometry-based methods as these technologies matured [85].

One new analysis method that came in the wake of genomic sequencing is *peptide mass fingerprinting* [95]. In this technique, protein samples are digested to peptides using proteolytic enzymes, and the masses of the intact peptides are measured with a mass spectrometer (using only a single MS stage). The peptide masses are then compared to theoretical peptide masses derived from the genomic sequences, and protein identifications can then be made based on a statistically significant number of peptide mass matches. However, this method too suffers from shortcomings that restrict its utility in large proteomics studies. If the analyzed sample is too complex (e.g., a mixture of a large number of proteins), or the sequence database being searched is too large, the protein assignments made with the peptide mass fingerprinting suffer from low statistical significance. Only after introducing a second stage of MS, which gave much needed information about the peptide's sequence and composition, did mass spectrometry start to become useful for high-throughput proteomics studies.

In the early days of MS/MS-based peptide identifications, a low volume of data and lack of effective algorithms made manual analysis the method of choice. The large growth in the volume of MS/MS data has forced researchers to rely on computational methods to perform the brunt of the analysis. There are three main computational approaches for peptide identification using MS/MS data: de novo sequencing, database search, and spectral libraries. Determining which approach to use depends on several factors such as the goals of the experiments (e.g., novel discoveries vs. routine diagnostics), the computational resources available, and the availability of supporting data

(such as genomic sequences).

1.D.1 De Novo Sequencing

De novo sequencing algorithms are designed to discover a peptide's amino acid sequence based solely on the information obtained from its experimental mass spectrum [9, 13, 18, 30, 49, 67, 69, 73, 74, 132, 134, 136, 146, 155, 185, 202, 221, 222, 245]. De novo algorithms search the space of all possible peptides in an attempt to find the one that best matches the mass spectrum's peaks. Early approaches attempted to explore this space by generating all possible peptide sequences that have a similar mass to the precursor mass of the experimental spectrum [183]. However, the exponential growth in the number of candidates that needed to be examined restricted this method to only very short peptides. Subsequent algorithms attempted to bound the search space by pruning candidates whose prefix did not have many supporting peaks in the spectrum [104, 109, 198, 238, 246]. More efficient approaches restrict the search space by modeling it as a graph [9, 13, 18, 30, 49, 69, 73, 120, 146, 221], and only consider peptides that are represented by paths in the graph. These graphs are typically scored using probabilistic models that evaluate how well the peaks in the observed mass spectrum correspond with the peptide's expected fragmentation pattern [9, 49, 69, 73, 136, 146].

Since de novo algorithms need no additional information beyond the mass spectrum itself, they are ideal for sequencing peptides when we do not have additional supporting information, such as the organism's genome. Even when genomic sequences are available, they are not necessarily accurate, since not all protein coding genes get correctly annotated. This is especially true for the many alternatively spliced genes, most of which are not adequately represented in the existing databases [127]. In such cases, many peptides that would not get identified in the traditional database search can still be sequenced using de novo methods.

The main drawback of using de novo sequencing algorithms is their low accuracy when sequencing low-resolution MS/MS spectra. Currently even the leading de novo algorithms correctly call only 70-75% of the amino acids [17, 69, 73], with only approximately 30% of all peptides identified without errors (as benchmarked for Lutefisk [221], SHERENGA [49], Peaks [136], NovoHMM [69] and PepNovo [73] (see Chapter 2). How-

ever, despite this low accuracy, de novo sequencing is useful for many applications, such as generation of peptide sequence tags for database filtration and homology based error-tolerant database searches (see Section 1.D.2 below), validation of database search results [233], construction of spectral networks [12], and shotgun protein sequencing [11].

Much of the work in this dissertation revolves around our de novo sequencing and its applications. In Chapter 2 we introduce PepNovo, our de novo sequencing algorithm and describe its novel scoring function. In Chapter 3 we demonstrate how PepNovo can be used to generate covering sets of peptide sequence tags that enable very efficient database filtration. The problems concerning de novo sequencing accuracy are mostly alleviated when it is performed on high-resolution data. In Chapter 4 we investigate how PepNovo can be used in these circumstances, and what new methods for peptide identification become feasible once de novo sequencing is more accurate. Finally, in Chapter 6 we develop a new powerful machine learning-based score, and demonstrate how it dramatically improves the performance of de novo sequencing.

1.D.2 Database Search

The most popular approach to peptide identification is the database search, which takes the query mass spectrum and scores it against a database of candidate peptides to detect significant matches. The lists of candidate peptides are generated by performing an in-silico digest of protein sequences (obtained from sets of annotated genes, open reading frames, etc.), and choosing all resulting peptide sequences with a mass that is similar to the precursor mass of the query spectrum. The first widely used program implementing this method was the Sequest algorithm [61, 236]. For each candidate peptide Sequest creates a theoretical spectrum that contains peaks for the expected fragments ions. The theoretical spectrum is then compared to the experimental one by computing their cross-correlation using Fast Fourier Transform [156]. Some of the other popular database search programs being used are Mascot [162], Spectrum Mill (by Agilent Technologies), ProbID [242], SONAR [68], Phenyx [39], OMSSA [84], X!-Tandem [44], and InPecT [219].

A lot of research has been devoted in recent years towards improving the performance of database search algorithms. One crucial aspect that has been receiving much

attention is the scoring used to evaluate peptide-spectrum matches, with researchers striving to develop more discerning functions [8, 25, 35, 39, 59, 84, 93, 96, 148, 162, 181, 182, 186, 207, 219, 229, 242]. In Chapter 6 we introduce a novel ranking-based scoring function that can significantly improve the sensitivity of these searches.

Database search programs have also been designed to give a better treatment of modified peptides that are a result of mutations from the database sequence or post translational modifications (PTMs) [94, 130, 133, 144, 160, 187, 219, 236], even when the types of modifications are not known in advance [168, 169, 217, 223].

There has been a lot of work devoted to speeding up the database searches. X!-Tandem [43, 44] uses a two pass search approach. It first rapidly identifies a list of candidate proteins, to which it restricts the second more time-consuming analysis(e.g., non-specific digestions or post-translational modifications, etc.) Many algorithms filter the database by using de novo methods to predict short sequence tags, and then only score peptides that match these tags [16, 33, 50, 75, 130, 139, 147, 197, 212, 219].

Often the protein sequences at our disposal do not adequately match the expressed proteome. This can occur when the protein samples come from unknown or unsequenced organisms, or even organisms with poorly annotated genes. In these cases, we can identify peptides by matching mass spectra to sequences of close homologues of the proteins being investigated. There are several error-tolerant sequence similarity-based algorithms that work along these lines. MS-Blast [194, 195] uses the popular similarity search algorithm Blast [4] to simultaneously align multiple de novo predictions with database sequences. MultiTag [208] performs a similar task but uses shorter peptide sequence tags. OpenSea [192] and Spider [89], align individual de novo sequences to a database one at a time, treating unaligned portions as either de novo sequencing errors, mutations, or post translational modifications.

One appealing property of database search algorithms is that their identifications can be validated and assigned confidence levels in the form of p-values or false discovery rates [56, 64, 65, 98, 112, 118, 121, 137, 149, 151, 175, 180]. One widely used validation method is the target-decoy strategy [10, 60, 97], where a shuffled version of the database is included in the search. By assessing the number of identifications made to the decoy database at different score levels, it is possible to derive score cutoffs based

on the estimated false discovery rates (FDRs).

1.D.3 Spectral Libraries

Tandem mass spectra of peptides are generally reproducible; repeated MS/MS experiments with the same sample generate very similar mass spectra (though they do exhibit some degree of variability due to the random nature of peptide fragmentation and slight differences in the conditions between repeated experiments [227]). However, in most cases the similarity between experimental spectra of the same peptide is so strong, one can confidently identify peptides by computing simple statistics such as the dot-product [204] or cross-correlation [61] between two spectra. This fact is the basis for a very accurate and powerful method of MS/MS identification using spectral libraries [46, 81, 125, 131, 204, 239]. Spectral libraries are simply collections of confidently identified spectra. Each query spectrum is compared to all spectra in the library with a similar mass in order to find a positive match; typically the number spectra in the library is orders of magnitude smaller than the number of peptides that would have to be considered in a database search.

Spectral libraries generally achieve higher accuracy and sensitivity rates than traditional database searches [46, 125], and also perform the identification much faster. However, spectral libraries are severely limited by the fact that they can only identify peptides that have sample spectra in the library. While spectral libraries can be continually augmented, they will still miss many identifications of uncommon peptides (PTMs, products of miscleavages, peptides from rarely expressed proteins or alternative splice variants, etc.)

1.E An overview of the dissertation

Most of the work we present revolves around algorithms for de novo sequencing of peptides, which aims to discover the amino acid sequence of protein digests (peptides), without relying on additional knowledge such as genomic sequences. We start off in Chapter 2 by describing a new scoring model which is used in our de novo sequencing algorithm called PepNovo. Our scoring method uses a probabilistic network whose

structure reflects the chemical and physical rules that govern the peptide fragmentation. We use a likelihood ratio hypothesis test to determine if the peaks observed in the mass spectrum are more likely to have been produced under our fragmentation model, than under a model that treats peaks as random events. We tested our de novo algorithm PepNovo on MS/MS data acquired on ion trap mass spectrometers, and achieved results that are superior to popular de novo peptide sequencing algorithms.

Filtration techniques in the form of rapid elimination of candidate sequences while retaining the true one are key ingredients of database searches in genomics. Although SEQUEST and Mascot perform a conceptually similar task to the tool BLAST, the key algorithmic idea of BLAST (filtration) was never implemented in these tools. As a result MS/MS protein identification tools are becoming too time-consuming for many applications including search for post-translationally modified peptides. Moreover, matching millions of spectra against all known proteins will soon make these tools too slow in the same way that "genome vs. genome" comparisons instantly made BLAST too slow. In Chapter 3 we describe the development of filters for MS/MS database searches that dramatically reduce the running time and effectively remove the bottlenecks in searching the huge space of protein modifications. Our approach, based on a probability model for determining the accuracy of sequence tags, achieves superior results compared to GutenTag [212], a popular tag generation algorithm.

The recent proliferation of novel mass spectrometers such as Fourier-Transform, Qtof and OrbiTrap marks a transition into the era of *precision mass spectrometry*, providing a two orders of magnitude boost to the mass resolution, as compared to low precision ion-trap detectors. In Chapter 4 we investigate peptide de novo sequencing by precision mass spectrometry and explore some of the differences when compared to analysis of low precision data. We demonstrate how the dramatically improved performance of de novo sequencing with precision mass spectrometry paves the way for novel approaches to peptide identification that are based on direct sequence lookups, rather than comparisons of spectra to a database. With the direct sequence lookup it is not only possible to search a database very efficiently, but it is also opens the possibility for using the database in novel ways, such as searching for products of alternative splicing or products of fusion proteins in cancer.

The analysis of the vast amounts of MS/MS data generated in recent years has raised formidable computational challenges. However, this data also opens a window of opportunity, making it possible to use advanced data-driven machine learning methods to devise more acute algorithms. In Chapter 5 we demonstrate how large MS/MS datasets can be used to learn some of the complex dynamics involved in peptide fragmentation. We use the RankBoost algorithm [77] to develop a discriminative ranking-based scoring function that predicts the ranks of peptide's fragment ion peaks in its experimental mass spectrum (the "peak rank prediction problem"). Our models are derived from simple sequence-based features (e.g., the identity of the amino acids in the vicinity of the fragmentation point). These features, each on their own right, are very weak predictors of the peak ranks. However, the RankBoost algorithms combines them together into a powerful and discriminating score.

In Chapter 6 we continue to exploit the large amounts of MS/MS data now available, and develop a new method for scoring peptide-spectrum matches that takes a discriminative boosting-based ranking approach, as opposed to the common use of generative statistical models. Our scoring models draw upon a large set of diverse feature functions that measure different qualities of peptide-spectrum matches (including features derived from our peak rank prediction models). We rely on RankBoost [77] to efficiently combine these features into a powerful and discriminating scoring function. Our new scoring model helps boost PepNovo's performance well beyond the state-of-the-art of de novo sequencing algorithms. Our score also greatly enhances the performance of database search programs. Since we are able to generate longer and more accurate sets of peptide sequences tags for database filtration, we are able to reduce InsPecT's search time by a factor of 15. In addition, our score also makes searches more sensitive, increasing the number of peptide identifications by 20%, in typical database searches. Our score proves to be especially valuable for proteogenomic searches in a six-frame translation of the human genome; a task poorly performed by current database search programs. Besides the great reduction in running time, which now makes these searches much more practical, we also see a large 60% increase in the number of identified peptides, compared to the number of peptides identified by InsPecT.

In Chapter 7 we examine how we can benefit from of the fact that tandem mass

spectrometry (MS/MS) experiments often generate redundant datasets containing multiple spectra of the same peptides. We develop a new clustering algorithm for MS/MS spectra that takes advantage of this redundancy by identifying multiple spectra of the same peptide, and for the purpose of analysis, replaces them with a single representative spectrum. Analyzing only representative spectra results in significant speed-up of MS/MS database searches. When analyzing large MS/MS datasets (over ten million spectra), clustering is a capable of reducing the number of spectra submitted to further analysis by an order of magnitude. The MS/MS database search of clustered spectra also results in fewer spurious hits to the database and increases number of peptide identifications as compared to regular non-clustered searches.

Recent advances in mass spectrometry instrumentation, such as FT-ICR and OrbiTrap, have made it possible to generate high resolution spectra of entire proteins. While these methods offer new opportunities for performing "top-down" studies of proteins, the computational tools for analyzing top-down data are still scarce. In Chapter 8 we investigate the application of *spectral alignment* [168, 169] to the problem of identifying protein forms in top-down mass spectra (i.e., identifying the modifications, mutations, insertions and deletions). We demonstrate how spectral alignment efficiently discovers protein forms even in the presence of numerous modifications, and how the algorithm can be extended to discover positional isomers from spectra of mixtures of isobaric protein forms.

2

The PepNovo Algorithm for De Novo Peptide Sequencing

We start off by describing PepNovo, our de novo sequencing algorithm. The key feature of our algorithm is a novel scoring function that uses a probabilistic network decomposition to create a more accurate model of peptide fragmentation. We tested our de novo algorithm PepNovo on MS/MS data acquired on ion trap mass spectrometers, and achieved results that are superior to popular de novo peptide sequencing algorithms.

2.A Background and Terminology

2.A.1 Fragment Ions

A peptide P is a sequence of n amino acids, $P = p_1 p_2 p_3 \dots p_n$, in an alphabet of 20 amino acids, each amino acid having a mass $m(p_i)$. The *precursor* of peptide P, is defined as $PM(P) = \sum_{i=1}^{n} m(p_i)+$ mass of H_2O. Generally in mass spectrometry experiments, peptides break along their backbones between successive amino acids during the stage of collision-induced dissociation (CID). This results in $n+1$ possible *cleavage sites* in a peptide (this count includes the empty peptide with mass 0, and the full peptide with mass PM).

A common event in the CID stage is a single cleavage along the peptide's backbone (see Section 1.C. Such a breakage results in a prefix fragment $p_1, \dots, p_i$ (also called

Table 2.1: Fragment ion statistics in low-resolution MS/MS proteomics data. Statistics collected from spectra in the OMICS [119] and OPD [171] datasets. The value of M used in the table depends on the type of fragment examined. When examining a prefix fragment, M is defined as the mass $M = \sum_{j=1}^{i} m(p_j)$, and when a suffix fragment is examined, M is defined as the mass $M = \sum_{j=i+1}^{n} m(p_j)$. The reported probabilities refer to the chance of observing expected fragments that have a mass in the "visible" portions of the spectra (for each spectrum this is the range of masses between the peak with the lowest mass and the peak with the highest mass). The probability for all expected fragment ions (for the whole range of masses) appears in parenthesis. The mass offsets in the table are rounded to the nearest integer value.

Prefix Fragments			Suffix Fragments		
Fragment Type	Offset	Probability	Fragment Type	Offset	Probability
b	$M+1$	0.83 (0.66)	y	$M+19$	0.87 (0.71)
$b-H_2O$	$M-17$	0.39 (0.30)	$y-H_2O$	$M+1$	0.26 (0.21)
$b-NH_3$	$M-16$	0.36 (0.28)	$y-NH_3$	$M+2$	0.24 (0.19)
$b-H_2O-H_2O$	$M-35$	0.13 (0.10)	$y-H_2O-H_2O$	$M-17$	0.11 (0.09)
$b-H_2O-NH_3$	$M-34$	0.12 (0.09)	$y-H_2O-NH_3$	$M-16$	0.13 (0.10)
b^{+2}	$(M+2)/2$	0.13 (0.08)	y^{+2}	$(M+20)/2$	0.23 (0.18)
$a \quad (b-CO)$	$M-27$	0.34 (0.26)			
$a-H_2O$	$M-45$	0.17 (0.13)			
$a-NH_3$	$M-44$	0.20 (0.15)			

an $N-$terminal fragment) and suffix fragment $p_{i+1}, \ldots, p_n$ (also called a $C-$terminal fragment). Since the original whole peptide, called the *precursor ion*, is charged, it is also possible for its fragments to retain a charge. Such charged fragments are also called *ion fragments*, and only they can be detected by a mass spectrometer. During the fragmentation process it is common for ion fragments to have *neutral losses*, which are chemical groups such as H_2O or NH_3 that get detached from the peptide fragments.

Table 2.1 lists some of the common fragment ions detected in low energy ion trap MS/MS which we chose to include in our model, along with their offsets from the cleavage site and the probabilities of detecting them in our data set. In typical ion trap mass spectra, ion fragment peaks are not detected in the low and high mass ranges. We therefore define the *visible* spectrum as the mass range in which intensity peaks appear, which corresponds to the masses between the spectrum's peak with the lowest mass and the spectrum's peak with the highest mass (in our data set the visible range covers 77% of an average spectrum). Table 2.1 reports the probabilities of detecting fragment ions both in the visible spectrum range, and in the entire spectrum range.

Our data is derived from doubly charged precursor ions, so the doubly charged ions b^{+2}, y^{+2} are possible fragments, and are included in our model. The fragments can be classified into two groups: prefix N-terminal fragments (b and a-ions, and their derivatives), and suffix C-terminal fragments (y-ions and their derivatives). If a cleavage of the peptide occurs at mass M between amino acids i and $i+1$, we can define the expected position for each of the fragment ions according to their offsets that appear in Table 2.1.

2.A.2 The De Novo Peptide Sequencing Problem and Spectrum Graphs

When the mass spectrometer is given a sample containing molecules of a peptide P, it fragments them using CID, and records the observed fragment masses and intensities in a mass spectrum S. This process can be viewed as drawing the spectrum S from the space of all mass spectra, according to a complex probability distribution $Prob(S|P)$, where $Prob(S|P)$ is governed by many factors such as the chemical composition of P, the mass spectrometer's properties, the experimental conditions, etc. The goal of sequencing algorithms is to find the peptide P that is the most likely source of S, i.e., the peptide P that maximizes $Prob(S|P)$ amongst all possible peptides. Since the distribution $Prob(S|P)$ is not available to us and is too complex to model, sequencing algorithms resort to using rough approximations in the form of scoring functions.

The space of all peptides is extremely large, making it inappropriate for an exhaustive case-by-case analysis. Database search algorithms reduce the size of the search space, by restricting their candidate peptides to ones that belong to the set of proteins present in the database. Most de novo algorithms restrict their search space to peptides that are paths in a spectrum graph. A *spectrum graph* [13, 49] is a directed acyclic graph; it's vertices correspond to putative prefix masses (cleavage sites) of the peptide. Two vertices are connected by a directed edge from the vertex with the lower mass to the one with a higher mass if the difference between them equals the mass of an amino acid. The Sherenga algorithm [49], uses a spectrum graph to sequence peptides by finding the highest scoring paths in the graph. The algorithm assumes that there is a set of k ion fragment types $\{y, b, y - H_2O, \ldots\}$, with a set of corresponding offsets from the cleavage site $\Delta = \{\delta_1, \ldots, \delta_k\}$. The vertices in the spectrum graph are assigned

by creating for each mass s_i in the experimental spectrum a set of k vertices at masses $s_i + \delta_1, \ldots, s_i + \delta_k$. Vertices $s_i + \delta_j$ and $s_{i'} + \delta_{j'}$ having similar masses are merged (since it is likely that they are created by different ion fragments from the same cleavage site). The vertices are scored according to a probability based score that gives premiums for present fragment ions, and penalties for missing ones.

2.A.3 Peak Offset Tolerance and Noise

The measurements reported by mass spectrometers are not always accurate. It is often the case that fragments are detected at slight offsets from their theoretical positions. In our scoring scheme we tolerate offsets of up to $\pm\varepsilon = 0.5$ Da of peak locations from their expected positions. The intensity of a fragment with expected mass x is determined by examining the peaks detected in the interval $[x - \varepsilon, x + \varepsilon]$ in the spectrum. Using $M = m$ as a putative cleavage site in the peptide, the fragment offsets define a set of intervals or bins $B_m = \{[b - \varepsilon, b + \varepsilon], [y - \varepsilon, y + \varepsilon], \ldots\}$, which correspond to the possible locations of the fragment peaks. Each interval in B_m is centered at its fragment's expected offset. For example, assuming $\varepsilon = 0.5$, we get that the bin for the b ion is $[m{+}0.5, m{+}1.5]$, the bin for the y ion is $[(PM{-}18){-}m{+}18.5, (PM{-}18){-}m{+}19.5]$, etc. When examining a cleavage at mass m our scoring method requires us to know the intensity levels for each of the possible fragment ions. We define the vector of ion fragment intensities $\vec{I} =< I_b, I_y, \ldots >$ to be the maximal intensity detected in each of the fragments' bins in B_m. A fragment bin that has no peak in it is given intensity 0.

Mass spectra typically contain many peaks for which there is no interpretation. In fact, in a typical mass spectrum most of the peaks are not annotated (though the majority of the high intensity peaks usually are - see Table 4.1). Some of these peaks are not annotated due to the limitations of our models. For example, they can belong to rare fragments (like x ions, or $a - H_2O - H_2O$, etc.). They can also be the result of complex events that are not covered by our model such as fragments from multiple cleavage sites on the same peptide (internal fragments). Another likely source for unannotated peaks are chemical contaminants and machine error. All these unannotated peaks are considered *noise* in the spectrum. The presence of many noisy peaks makes de novo sequencing difficult, since the noisy peaks can cause random false matches with ion fragments. In

our data, the probability that a random peak matches an ion fragment's position is approximately 0.1 (for the visible region of the spectra). Though it might seem that this means that some of the low probability ion fragments mentioned in Table 2.1 are not distinguishable from noise (e.g. $y-H_2O-H_2O$, $b-H_2O-H_2O$, etc.), this not always the case. As explained below, there are situations in which these fragments contribute to the score, such as when we consider them in a combination with other ion fragments, or when they appear in sparse regions of the spectrum.

2.A.4 Discretizing Intensities and Cleavage Positions

Our scoring model considers two types of continuous values that need to be transformed into discrete values, which are more convenient to use with our models. Spectrum peak intensities are assigned k discrete *intensity levels* using experimentally derived thresholds. Depending on the experimental conditions, spectra can have total intensities that span several orders of magnitude. We therefore assigned the peaks relative intensity levels (rather than using fixed thresholds). This is done by calculating a baseline *grass* intensity for each spectrum, which equals the average of the intensities of the weakest 33% of the peaks in the spectrum. We then divide each peak's intensity by the grass level, to determine it's normalized intensity. Using the training data, we experimented with several different numbers of intensity levels, and different threshold values to separate between intensity levels. Let I denote the normalized intensity level of a peak; we obtained optimal results using the following 4 intensity levels: 0 (*zero*) is assigned to peaks with $I < 0.05$, 1 (*low*) is assigned to peaks with $0.05 \leq I < 2$ (62% of the peaks in the training data), 2 (*medium*) is assigned to peaks with $2 \leq I < 10$ (26% of the peaks), and 3 (*high*) is assigned to peaks with $I \geq 10$ (12% of the peaks).

The other type of value discretized in our models is the relative position of a cleavage site m in a peptide of mass PM. The relative position is defined as $pos(m) = \frac{m}{PM}$. We discretized the values of $pos(m)$ into $k = 5$ equally sized regions labeled $0, \ldots, k-1$, i.e., $pos(m) = 0$ denotes a cleavage in the first fifth of the peptide near the $N-$terminal, $pos(m) = 1$ denotes a cleavage in the second fifth, etc. We added this variable to our model because the intensity of observed peaks is correlated with the region in the peptide in which the peaks appears. On average, peaks are stronger in the

center of the peptide, and are weak or missing near the terminal ends.

2.B The New Likelihood Scoring Method

In this section we propose a scoring scheme that assigns a relevance score to peptide prefix masses (which are the vertices of the spectrum graph). For each mass m our scoring function determines how likely it is that there was a cleavage of a peptide at mass m, i.e., that m is the mass of a prefix of the peptide P that created the spectrum S.

2.B.1 The Hypothesis Test

At the heart of our scoring function is a hypothesis test. Hypothesis tests are used by several existing scoring functions [25, 37, 49, 59, 93]. Our hypothesis test compares between two competing hypotheses regarding a spectrum S and a mass m of a possible cleavage site (these hypotheses are expressed using statistical models). The first hypothesis is the Collision Induced Dissociation model (CID) which states that m is a genuine cleavage in the peptide that created S. According to this hypothesis, there are rules that govern the outcome of peptide fragmentation. In particular, there are certain combinations of fragments and intensities that are more probable than others. We use a probabilistic network that models these fragmentation rules to determine the probability $P_{CID}(\vec{I}|m,S)$ of detecting an observed set of fragment intensities $\vec{I}$, given that mass m is a cleavage site in the peptide that created S. The competing hypothesis is a random peaks hypothesis ($RAND$), which assumes that the peaks in the spectrum are caused by a random process. Thus, the intensities that appear in $\vec{I}$, that supposedly belong to fragment ions due to a cleavage at mass m, are in fact only random peaks that happen to fall into the designated bins. We describe how to compute the probability $P_{RAND}(\vec{I}|m,S)$ in this scenario.

The score given to a mass m and spectrum S is the logarithm of the likelihood ratio of these two hypotheses,

$$Score(m,S) = \log \frac{P_{CID}(\vec{I}|m,S)}{P_{RAND}(\vec{I}|m,S)}. \tag{2.1}$$

A positive score in Eq.(2.1) means that according to our models, it is more likely that the peak intensities $\vec{I}$ were caused by a genuine cleavage event (the higher the score, the likelier this hypothesis is, compared to the competing random hypothesis). Likewise, a negative score means that the observed intensities $\vec{I}$ are probably due to random peaks, and they give no credence to a cleavage at mass m. We now describe in detail how to compute the probabilities $P_{CID}(\vec{I}|m,S)$ and $P_{RAND}(\vec{I}|m,S)$, under these two different hypotheses.

2.B.2 The Collision Induced Dissociation Hypothesis

According to the CID hypothesis, there are rules that govern the outcome of the peptide's fragmentation process in the mass spectrometer. These rules define which ion fragments and which peaks intensities are more likely to be observed. We chose to include in our CID model three types of factors that are a result of mass spectrometry fragmentation rules:

1. Dependencies and correlations between types of fragment ions.
2. The positional influence of the cleavage site (the influence of the relative region in which the fragmentation occurs).
3. The influence of the the type of amino acids directly N-terminal and C-terminal to the proposed cleavage site.

At this stage, we restrict our discussion of the scoring method to include only the first two factors mentioned above. The incorporation of the effect of the flanking amino acids to the cleavage site is treated separately in a section below.

Figure 2.1 illustrates the probabilistic network (described by a directed acyclic graph) that we use to model the common fragments resulting from a peptide cleavage. A vertex u in the graph is called a parent of vertex v if the there is a directed edge (u,v) in the graph. There are three vertices in our graph without parents, two which involve the amino acids flanking the cleavage site (the vertices N-aa and C-aa), and the vertex $pos(m)$ which holds the relative region in the peptide in which the cleavage occurs. All other vertices are labeled with the fragment types from Table 2.1. Each of

these vertices holds a conditional probability table of the value of the vertex given the values of its parents. For instance the vertex b holds a table which gives the probability $P(b = I_b|y = I_y, pos(m) = r)$, where I_b is the intensity detected for the b-ion, I_y is the intensity detected for the y-ion, and the cleavage occurred in the peptide at region r.

The values in the tables were filled by empirically counting in the training data the number of appearances of each possible combination of variables in the table (we had a training set of 972 spectra from which we drew statistics, see Section 2.C). Some variable combinations did not appear, resulting in zero counts. We smoothed these zero counts by adding a small uniform count to all combinations.

After exploring several network configurations which included different types of fragments, and various ways to connect between them, we found that the structure depicted in Figure 2.1 gives the best results. Note that this structure reflects fragmentation rules that arise from our training data which consists of spectra of doubly charged tryptic peptides obtained from an ion trap mass spectrometer. Spectra from other types of mass spectrometers, charge states, or proteolytic enzymes can lead to significantly different fragmentation rules.

The edges that appear in the graph reflect two types of dependencies and causal relations (at this stage we ignore edges emitting from the vertices N-aa, C-aa). The first type of dependencies modelled are the correlations between the intensity levels of the ion fragments. Though to some extent there is a correlation between all ion types, some combinations tend to display higher correlation in their intensities. For instance, the b and y ions are highly correlated. In ion trap data, when a cleavage in a doubly charged tryptic peptide creates a high intensity y-ion, there is usually also a high intensity b-ion. We model this phenomenon by adding an edge between the vertex y and the vertex b (the direction of the edge in this case is arbitrary). The extent of this dependence can be seen when we examine the probability tables. For instance, the probability of seeing a strong b-ion in the center of the peptide, given that there is a strong y-ion, is $P_{CID}(I_b = high|I_y = high, pos(m) = 2) = 0.36$, and it drops to 0.03, if instead of the strong y-ion, a weak y-ion is detected ($I_y = low$). This large difference in probabilities is due to the fact that in spectra of tryptic peptides, the y-ions are usually stronger than their b counterparts [212]. It is therefore unlikely to detect a case where the b-ion is much

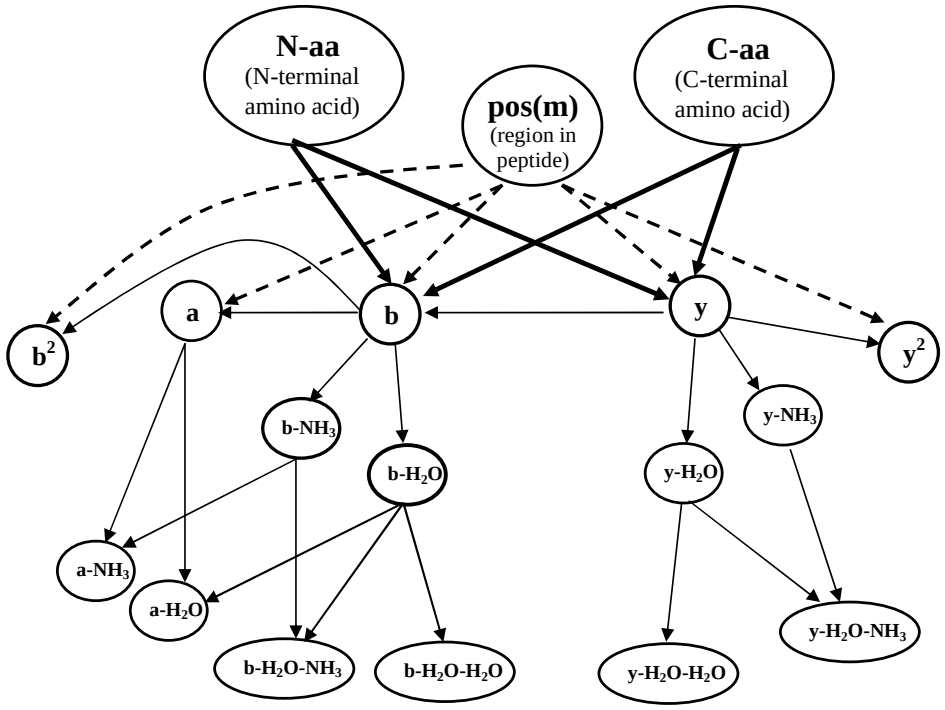


Figure 2.1: The probabilistic network for the CID fragmentation model of doubly charged tryptic peptides measured in an ion trap mass spectrometer. Three different types of relations are modelled in this network: (1) correlations between fragment ions (regular arrows); (2) dependencies due to the relative position of the cleavage site in the peptide (dashed arrows); (3) influence of flanking amino acids to the cleavage site (bold arrows).

stronger than the y-ion. Dependencies of this type were not accounted for in previous de novo scoring models and adding them to our score led to improved performance.

There are also correlations between the intensities of ion fragments and their neutral losses. For instance, if we do not detect a b-ion, we are less likely to detect a $b - H_20$ ion. This is reflected in the probability tables by the values $P_{CID}(I_{b-H_2O} > zero|I_b = high) = 0.496$ for detecting a $b-H_2O$ ion when a strong b-ion was also detected, compared to the probability $P_{CID}(I_{b-H_2O} > zero|I_b = zero) = 0.242$ of detecting a $b - H_2O$ ion when no b-ion was detected. Not all the correlations between ion fragments have edges in our model's graph. For instance, a strong y-ion can indicate that the intensity of other prefix fragments will also be high (besides the b ion which is correlated with y). In this case, it might be reasonable to add edges from y to other prefix fragments,

however the information about y can be mediated quite well by the value of b (since a strong y is likely to be coupled with a strong b). In the interest of simplifying our model we chose not to add those edges.

The second type of dependency modelled in our graph is the effect of the region in which the cleavage occurs (the vertex $pos(m)$). There are edges from the vertex $pos(m)$ to the vertices b, y, a, y^{+2} and b^{+2}, because the intensity of these fragments depends on where in the peptide the cleavage occurred. For instance, y and b-ions tend to have higher intensities in the middle of the peptide, whereas they are hardly detected near its ends [93]. a-ions tend to be detected more when cleavages occur in the first half of the peptide. Since it is more likely for larger fragments to retain both charges in doubly charged peptides, the b^{+2} ions are observed more often when the cleavage occurs towards the C-terminal, whereas the y^{+2} ions are observed more often when the cleavage is closer to the N-terminal. Of course the cleavage location also has a strong influence on the rest of the fragment ions, but for the benefit of a simpler model, we chose to omit these edges.

The reason it is beneficial to simplify probabilistic networks becomes clear when we examine how the model complexity is affected by the addition of an edge. Each additional edge that points to a vertex adds a dimension to the probability table of that vertex. Assuming there are x edges entering a vertex, and there are k discrete intensity levels, the size of the probability table at the vertex is k^{x+1}. Since we have a limited number of training spectra, if we do not restrict the model's complexity, *over-fitting* will occur. When this happens, the model's parameters are too biased towards fitting the training data and do not generalize well to accommodate data that is different from the samples in the training set.

We use the probabilistic network of Figure 2.1 to compute $P_{CID}(\vec{I}|m, S)$, the probability of observing ion fragment intensities $\vec{I}$ given that the putative cleavage occurred at mass m in spectra S. We denote by $V = \{b, y, \ldots\}$ the vertices in the probabilistic network, excluding the vertices $pos(m)$, N-aa and C-aa. For each vertex $v \in V$, $\pi(v)$ denotes the set of v's parents in the graph ($\pi(v)$ are the vertices that have edges pointed from them to v), and $\vec{\pi}(v)$ denotes the set of values assigned to the vertices $\pi(v)$. $P_{CID}(I_v = i|\vec{\pi}(v) = \{i_1, i_2, \ldots\})$ is the probability of detecting the intensity i at

fragment ion v given the intensities detected at its parents. According to the properties of this type of probabilistic network, a vertex v is independent of the other vertices in the graph given that the values of its parents are known (this network is a casual network with all the vertices instantiated [107]). This leads to the following decomposition for the probability of the intensities $\vec{I}$

$$P_{CID}(\vec{I}|m,S) = \prod_{v \in V} P_{CID}(I_v|\vec{\pi}(v),m,S) \tag{2.2}$$

Since the values in the conditional probability tables in our model were derived from true mass spectrometry data and represent some of the rules governing the fragmentation process, the probability P_{CID} can help distinguish between likely combinations of ions (that are frequent in real cleavage sites), and unlikely combinations. For instance, the probability assigned to instances where both ions and their neutral losses are detected should be higher than unlikely instances such as ion combinations where neutral losses are detected without the b or y-ions registering any intensity.

The fact that our model considers combinations of ion fragments makes it possible, in certain situations, for low probability fragments to contribute to the scoring. This happens because our model considers the fragment's intensity in context with other fragments, and can identify combinations that have a probability that deviates from the random background probability. For instance, the average probability of detecting a $y-H_2O-H_2O$ ion fragment is 0.11 (see Table 2.1), and thus should be virtually indistinguishable from random noise peaks (that have probability 0.1). However, when it is considered together with the $y-H_2O$ fragment, there are combinations for which the probability of detecting the $y-H_2O-H_2O$ is higher, for example $P(y-H_2O-H_2O = medium \,|\, y-H_2O = high) = 0.17$, and could thus make a positive contribution to the score.

2.B.3 The Random Peak Hypothesis

The random model assumes that peaks are distributed according to some simple prior distribution throughout the spectrum, without there being any special cleavage sites or fragmentation rules that influence the detection of peaks at certain offsets. When we observe the intensities of $\vec{I}$ from a cleavage at mass m , any peak matches with

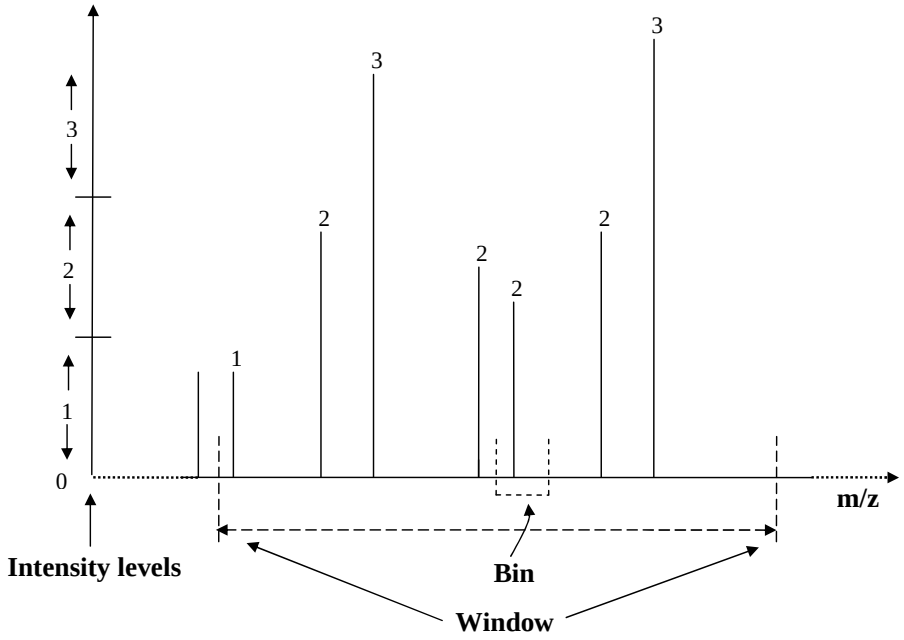


Figure 2.2: Counting peaks in a window placed around a bin. There are $n_1 = 1$, $n_2 = 4$, and $n_3 = 2$, peaks of the respective intensity levels in the window. The designated bin contains a single peak with intensity 2.

fragment bins are considered to be due to chance. Under this random model, each peak is distributed independently of the others. Thus the probability $P_{RAND}(\vec{I}|m, S)$, can be computed as the product of the probabilities of seeing the individual peaks in their bins.

To compute the probability of randomly seeing a peak with intensity level t in a bin of width 2ε around mass m', we use an empirical estimate of the peak density in the vicinity of m'. This local density estimation is used because peaks are not distributed uniformly throughout the spectrum mass range. For instance, peaks tend to be stronger and denser towards the center of the spectrum, and sparser and weaker near the terminal ends. The density estimation is done by looking at a window of width w around the mass m', and counting how many peaks of each intensity level i appear in this window. Assuming there are d different intensity levels, we denote these counts by n_i, $1 \leq i \leq d$. Figure 2.2 illustrates such a count.

Let $\alpha = 1 - \frac{2\varepsilon}{w}$ be the probability of uniformly selecting a random location for a single peak in a window w, and having it fall outside a specified bin of width 2ε. The

probability that the highest intensity level for a peak detected in a bin centered at m' is $t \geq 1$, given the peaks counts $n_1, \ldots, n_d$ in a window of width w around m', is given by the following equation

$$P_{RAND}(I = t|n_1, n_2, \ldots, n_d) = (1 - \alpha^{n_t}) \cdot \alpha^{\sum_{i=t+1}^{d} n_i} \tag{2.3}$$

Eq.(2.3) can be explained as follows. If the maximal intensity in the bin is t, we want to avoid the case where all the peaks of intensity t in the window w miss the bin (we don't mind if several peaks with intensity t or lower happen to also fall in the bin). The probability that a random placement of all peaks with intensity t misses the bin is α^{n_t}, so the complimentary event where at least one peak with intensity t falls in the bin has probability $1 - \alpha^{n_t}$. As for the higher intensity peaks, we want them all to miss the bin, and the probability that that occurs is $\alpha^{\sum_{i=t+1}^{d} n_i}$. Following this reasoning, the probability that no peak falls in a bin is given by

$$P_{RAND}(I = 0|n_1, n_2, \ldots, n_d) = \alpha^{\sum_{i=1}^{d} n_i} \tag{2.4}$$

Eq. (2.3) together with Eq.(2.4) define a probability density function for which

$$\sum_{i=0}^{d} P_{RAND}(I = i|n_1, n_2, \ldots, n_d) = 1 \tag{2.5}$$

We assume that in the random model, the events of detecting peaks in bins are independent of each other. Therefore, we can factor the probability $P_{RAND}(\vec{I}|m, S)$ of detecting a combination of our model's k fragments' intensities into the product of the individual probabilities, as follows

$$P_{RAND}(\vec{I}|m, S) = \prod_{i=1}^{k} P_{RAND}(I_i|n_{i1}, n_{i2}, \ldots, n_{id}) \tag{2.6}$$

By examining Eq. (2.3), we can gain insight on how the random model helps to balance the effects of noise. When many noisy peaks are present (typically having low intensity), they can cause random matches, and thus supposedly increase the score for a cleavage site. However, if we look at Eq. (2.3), we see that increasing the peak count for the low intensity peaks also increases the probability of detecting such a peak by chance. Since the probabilities of the random model appear in the denominator of the score equation (see Eq. 2.1), the result is a decrease in the score. Thus, if an ion

fragment is detected in a dense region of the spectrum, it contributes less to the score compared to the contribution it would bring had there been only a few peaks in it's vicinity. This correction does not occur when a simple random model is used, such as the one used by Dancìk et al. [49], where the same constant random probability is used for all regions in the spectrum.

2.B.4 Modeling The Influence of Flanking Amino Acids

Recent research has uncovered many chemical properties and pathways that influence the outcome of the CID fragmentation process. It has been suggested that incorporating such information can improve scoring function [101, 191, 212]. A recent scoring function that uses this type of information obtained high accuracy for database searches [59], however incorporating such information into de novo sequencing algorithms has been an open problem.

An amino acid is said to have an N-terminal bias if on average, the b and y peaks at the cleavage site $N-$terminal to the amino acid are stronger than the peaks from the cleavage on its $C-$terminal side. Similarly, an amino acid exhibits $C-$terminal bias if the average intensity of peaks from the cleavage C-terminal to the amino acid are stronger than the $N-$terminal ones. Some of the prominent amino acid biases and preferred cleavage sites that have been mentioned in the literature are:

- N-terminal bias of proline, glycine and serine [23, 212].
- C-terminal bias of aspartic acid [86] (especially in proteins with no mobile protons [234]).
- Influence of histidine on cleavage C-terminal to acidic residues [103].

A qualitative measurement of some of the aforementioned phenomena is given in [102, 212]. Some of these biases are very strong, for instance the b and y peaks N-terminal to Proline are typically at least 5 times stronger than their $C-$terminal counterparts. Adding this information into the model can help to determine genuine cleavage sites.

We incorporate the amino acid biases into our model by adding the vertices N-aa and C-aa (see Figure 2.1), and adding directed edges from them to the vertices b

and y. These edges add two conditioning variables to the conditional probability tables for b and y. Since there are 20 different amino acids, adding these variables makes the conditional probability tables for b and y 400 times larger. This large increase in table size requires much more training data than we have available to us. To reduce the number of parameters needed to train, we grouped the different amino acid combinations into 16 equivalence sets. The assignment of amino acid pairs to equivalence sets is done according the order rank of the sets, i.e., any two amino acids are assigned to the highest ranking set to which they can belong. The equivalence sets we use are as follows (X- denotes any amino acid, we start our list from the highest ranked set): X-Pro, Pro-X, X-Gly, Gly-X, X-Arg/Lys, His-X, X-His, Asp/Glu-X, X-Asp/Glu, Ile/Leu/Val-X, X-Ile/Leu/Val, Ser/Thr-X, X-Ser/Thr, Asn-X, X-Asn, X-X (any combination of two amino acids). If both amino acids in the pair are either glycine or proline, we assign the combination to the X-X set (since there is less cleavage in these cases [102]). The sets' order was determined according to the extent each amino acid influences the intensities of the peaks at a cleavage site and causes a deviation from the typical cleavage intensities (we determined this based on the results mentioned in refs. [102, 191]). For instance, in most cases, proline and glycine have a stronger influence than the other amino acids, therefore they are placed at the top of the list. Note that by using such equivalence sets, we actually model the influence of only one of the flanking amino acids each time, though it is usually the dominant one (since the sets with the dominant amino acids appear higher in our ranking). A more accurate approach might be to model the contribution of both flanking amino acids [191], however as mentioned above, this requires a larger training set than the one that was available to us.

Using the expanded conditional probability tables, we can replace the probability $P_{CID}(\vec{I}|m,S)$ of Eq.(2.2), with $P_{CID}(\vec{I}|m,S,N{-}aa,C{-}aa)$. Note that adding the conditioning on the N and C-terminal amino acids only affects the probabilities of the fragments b and y. The other fragments' tables are not affected by this, for instance $P_{CID}(I_{y-H_20}=i_1|I_y=i_2,N{-}aa,C{-}aa)=P_{CID}(I_{y-H_20}=i_1|I_y=i_2)$. Furthermore, there is no need to make any changes to the random model because of the added conditioning on the flanking amino acids, since it is assumed in that model that the peaks are created in a random process which is not governed by the fragmentation of any source

peptide.

The addition of the $N-$aa and $C-$aa vertices to our model changes the way we score vertices in the spectrum graph. Before the addition, each vertex in the spectrum graph had a single score. Now, each vertex can have 16 different scores (for the different combinations of flanking amino acids). When searching for the high scoring path, the de novo algorithm must select for each vertex its appropriate score, depending on the edges which enter and exit the vertex.

2.B.5 The PepNovo De Novo Sequencing Algorithm

When given a query spectrum, the first step in the de novo sequencing algorithm does to construct a spectrum graph. The vertices in the spectrum graph represent possible cleavage sites, and the solution interpretations correspond to high scoring paths in the graph. For this reason, selecting the appropriate number of vertices for the spectrum graph is essential for obtaining optimal results. On the one hand, if too few vertices are selected many cleavage sites can be missed, and the graph might contain several disconnected sub-paths of the correct solution. On the other hand, if too many vertices are used, this causes many spurious edges, which create high scoring incorrect sup-paths that add noise which masks the correct path.

Our method for determining the graph's vertices is as follows. Given a query spectrum S, we first select part of the peaks in the spectrum, choosing only the strongest peaks in each region. This is done by sliding a window of width w across the spectrum and keeping any peaks that are in the top k, for some window location. For $w = 56$ Da and $k = 3$, this selects on average 62 peaks per spectrum, which is a density of 5.2 peaks for every 100 Da. Since the highest peaks in the spectrum tend to be b and y-ions, we create vertices for both of these interpretations: Given a peak at mass x, we create a vertex at mass $m = x - 1$ (by interpreting the peak as a b-ion) and also create a vertex at mass $m = PM - x + 1$ (by interpreting the peak as a y-ion). To these vertices we add the vertices for mass 0 (the empty peptide), and mass $PM - 18$ (which is the mass of the entire peptide). We merge vertices that are within 0.5 Da of each other (since they are likely created from b and y ions of the same cleavage site). When following this procedure, the average number of vertices in a spectrum graph for the test set is

110. Note that this method is different from the method used by Dancik *et al.*, where all peaks (and all their interpretations) where used to select the vertices in the spectrum graph. The edges in the graph are created by connecting vertices that have a mass which approximately equals the mass of an amino acid (we used a tolerance of ± 0.5 Da).

Scoring vertices in the spectrum graph is done by taking each vertex's mass m and finding the intensities $\vec{I}$ of the fragment ions for a cleavage at mass m in the original spectrum S (containing all peaks). We then score the vertex according to the log-likelihood score of Eq. (2.1). Note that each vertex has several scores computed for it according to the different combinations of flanking amino acids. When performing its search for a high scoring path, our search algorithm selects the appropriate score for the vertex, according to the combination of edges it uses in the path that goes through that vertex.

It is common in mass spectra for peaks to have *isotopic* peaks that appear at increments of one Dalton after the peak. The isotopic peaks are caused by peptide fragments that contain isotopic atoms (the most common is isotope ^{13}C but N, O, and S can also contribute to this). Isotopic peaks are usually detected for strong peaks, therefore it is common for the b and y ions to have additional peaks at offsets of $+1$ and $+2$ Da. These isotopic peaks can create additional vertices in the spectrum graph which can lead to sequencing errors. One approach to deal with isotopic peaks is to remove their vertices from the graph. This however can lead to the removal of genuine vertices, that were created from peaks that happen to fall in the isotopic peak positions. Instead of using this approach, we chose to give vertices a premium to the score if their b or y peaks had isotopic peaks ahead of them, and give the vertices a score penalty if their b or y peaks seemed to be isotopic peaks themselves (that is they had strong peaks at an offset of -1 Dalton).

A point that needs to be kept in mind when constructing spectrum graphs, is that the experimental precursor mass measured in mass spectra machines is often inaccurate, and can thus lead to mistakes in the de novo sequencing. To solve this problem we use the combinatorial parent mass correction procedure from by Dancik *et al.* [49].

Once the spectrum graph is created and scored, we need to find a highest scoring

antisymmetric path in it [49]. Since every peak we use from the spectrum contributes two vertices to the spectrum graph, we could end up with symmetric paths which use both vertices attributed to a peak. This leads to incorrect interpretations. Therefore, we restrict our solutions to paths containing at most one vertex from each of these "forbidden pairs" of vertices. Though this problem is generally intractable, the unique structure of the forbidden pairs in the spectrum graphs leads to a polynomial time algorithm for the antisymmetric path problem [30]. In order to find the highest scoring path in the graph, we used a dynamic programming algorithm similar to the one due to Chen *et al.* [30, 132] that is modified to take into account the particulars of our scoring function (see section on constructing the spectrum graph). Our dynamic programming algorithm is also capable of returning sub-optimal paths, if desired by the user.

2.C Experimental Results

Mass Spectra Data Set

The dataset we used is composed of doubly charged tryptic peptides obtained from low energy ion trap LC/MS/MS runs. We limited our experiments to only dealing with spectra of doubly charged precursor ions since this charge state is the most common in many mass spectrometry experiments. In total we obtained 1252 spectra of peptides with unique sequences which were identified with high confidence by Sequest (these spectra had $Xcorr > 2.5$ and came from proteins with multiple hits). Our data came from two sources, the ISB protein mixture dataset [118] which used an ESI-ITMS mass spectrometer made by ThermoFinnigan, San Jose, CA, and the Open Proteomics Database (OPD) [171], which used a ESI-Ion Trap "Dexa XP Plus" mass spectrometer, also from ThermoFinnigan. Ideally a scoring model should be trained using spectra from a single type of machine. However, in order to create a sufficiently large training set, we resorted to using spectra from these two different sources (although both are ESI-ion trap instruments that produced spectra with similar characteristics). For our test set, we selected from 280 spectra from this data set, corresponding to peptides with an average length of 10.7 amino acids.

Measuring Accuracy of De Novo Predictions

We desired a metric by which the success of de novo reconstructions could be evaluated and compared with other algorithms. Since all benchmarked algorithms produce a single de novo prediction, the natural parameter we can look at is the *prediction accuracy*, which is defined as:

$$Prediction\ Accuracy = \frac{\#\ correct\ amino\ acids}{\#\ number\ of\ predicted\ amino\ acids}. \tag{2.7}$$

However, de novo sequencing algorithms can also predict partial, rather than complete peptides, so a high score on this parameter can be obtained by only predicting high scoring short partial peptides. Usually, this includes the portion in the center of the peptide that has stronger peaks, while amino acids near the terminals are ignored. We therefore also look at the capability of the algorithms to reconstruct correct consecutive subsequences of amino acids (that appear in the prediction in their expected position according to the correct peptide). For each prediction made by the algorithms, we determined the maximal correct subsequence, and tallied the counts for the entire test set. Note that a predicted amino acid (or subsequence) is considered correct if its position in the predicted de novo sequence is within 2.5 Da from its expected position according to the correct sequence. We use this large margin to account for offsets in amino acid locations that occur both due to inaccurate peak m/z measurements, and an incorrect precursor mass (even after precursor mass correction is used). In addition, we do not make a distinction between the amino acids leucine and isoleucine (which have identical masses), and between lysine and glutamine (which have a small difference of 0.04 Da in their masses).

Benchmark Results

We compared the performance of PepNovo with the following popular de novo sequencing algorithms: Lutefisk XP v1.0 [222], Peaks v2.4[136], and Sherenga [49] (which is included in the Spectrum Mill v3.01 software suite).

We ran the algorithms on each of the 280 test spectra, and kept the highest scoring interpretation they returned. The following parameters and settings were used for this benchmark. Lutefisk was run with the default parameters for doubly charged

Table 2.2: Comparison of de Novo peptide sequencing algorithms. The table holds cumulative results for 280 test spectra: the average accuracy of predicted amino acids, average prediction length, and the proportions of predictions that had a correct subsequence of length at least x, for $3 \leq x \leq 10$.

Algorithm	Average Accuracy	Average Length	Predictions With Correct Subsequences of Length at Least x							
			$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x = 10$
PepNovo	**0.727**	10.30	**0.946**	**0.871**	**0.800**	**0.654**	**0.525**	**0.411**	0.271	**0.193**
Sherenga	0.690	8.65	0.821	0.711	0.564	0.364	0.279	0.207	0.121	0.071
Peaks	0.673	10.32	0.889	0.814	0.689	0.575	0.482	0.371	**0.275**	0.179
Lutefisk	0.566	8.79	0.661	0.521	0.425	0.339	0.268	0.200	0.104	0.057

tryptic peptides on ion trap mass spectrometers. Peaks was run with an error tolerance of 0.6 Da, trypsin digestion, and treating Q/K and I/L as identical amino acids. Sherenga was run with ESI ion trap scoring, minimum vertex score 0, and treating I/L and Q/K as identical amino acids.

The results of the four de novo algorithms are given in Table 2.2. PepNovo, Peaks and Sherenga all achieve results superior to Lutefisk's, both in terms of accuracy, and in terms of the longest correct subsequences predicted. As far as the prediction accuracy is concerned, PepNovo has the highest accuracy even though on average Sherenga makes shorter predictions and thus has an advantage since it is making more selective predictions (this enables it to get a higher accuracy than Peaks). When we examine the prediction of correct consecutive amino acid sequences, we see that PepNovo obtained the best results, with Peaks coming in a close second, especially when the longer subsequences are concerned.

We also ran additional experiments with deficient versions of PepNovo, where each variant of the algorithm was lacking one of the components that are incorporated into the PepNovo scoring model (e.g. dependencies between fragments, information on flanking amino acids, intensity thresholds, etc.) The results are given in the supporting information. Each of the tested components proved to have a positive influence on PepNovo's performance (since all deficient versions of PepNovo had inferior success rates). For instance, a version of PepNovo that did not use information on the flanking amino acids showed a reduction of 1.5% to the prediction accuracy. It is likely that the improvement in performance due to adding flanking amino acids to the model would be greater than 1.5% if more training data were available, enabling the inclusion of more

equivalence sets, possibly to the degree of having a separate probability table for each pair of flanking amino acids. The lack of other components in the model, such not having intensity thresholds or using a simple random model, caused a larger decrease in the performance (see table in supporting information for more details). We also evaluated our de novo algorithm with Dancik scoring (which lacks many of PepNovo's enhancements), and found that PepNovo's scoring performs much better both in terms of the prediction accuracy (70.4% vs. 59.2%) and in terms of the counts of the maximal lengths of correct subsequences in the predictions.

2.D Discussion

The results obtained for PepNovo demonstrate the power of our new scoring model, which enabled our algorithm to outperform popular de novo algorithms. After the publication of the PepNovo algorithm [73], two additional de novo algorithms were published benchmark experiments on our dataset. NovoHMM [69] achieved similar results to PepNovo: an amino acid prediction accuracy of 73.6% vs. PepNovo's 72.7%, but had lower percentages of correct subsequences (e.g., 91.1% of the predictions contained a correct subsequence of length 3 vs. 94.6% for PepNovo). The second algorithm published is MS-Novo [146] which boasted superior performance to the other existing algorithms, however when applying the same benchmarking methodology used above, the actual performance of MSNovo was well below that of PepNovo, Peaks and NovoHMM.

The PepNovo program is very efficient, typical running time is less than 0.1 per spectrum (much faster than the other algorithms which typically require 1 second or so per spectrum). This makes it useful for other tasks besides de novo sequencing. In Chapter 3 we explore how PepNovo can be used to generate peptide sequence tags for database filtration. PepNovo's scored spectrum graphs have been incorporated into the building of spectrum networks [11, 12] and PepNovo's de novo predictions have been used for verification of marginal database hits [233], and the identification of proteins from unsequenced organisms [230] using the MS-Blast algorithm [194].

One area in which PepNovo can be greatly enhanced is the scoring models. We revisit this problem in Chapter 6 where we present a new machine learning method which

incorporates much more of the mass spectrometry "wisdom" into the scoring process, and delivers superior de novo sequencing performance. Another aspect that is not addressed in PepNovo is the reliability of the results. In Section 3.B.2, we examine a method for assessing the accuracy of the de novo predictions, and in particular assigning reliability scores to individual amino acids in the predictions.

This chapter, in full, was published as "PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling". A. Frank and P. Pevzner. *Analytical Chemistry*, 77:964-973, 2005. The dissertation author was the primary author of this paper.

3

Peptide Sequence Tags for Database Filtration

3.A Introduction

Although MS/MS database search algorithms such as Sequest [61] and Mascot [162] offer high-throughput peptide identification, these algorithms do not give a complete solution to this problem, particularly in the case of spectra from *Post-Translationally Modified* (PTM) peptides. The requirement to consider every modification over every possible peptide, not only makes the process too slow, but also makes it harder to distinguish the true peptide from false hits. As a result, the mass-spectrometrists currently face a challenging computational problem: given a large collection of spectra, find out which modifications are present in each protein in the sample.

In a sense, the protein identification problem is similar to one faced by the genomics community in their search for sequence similarities. The solution is to use *filters* that quickly eliminate much of the database, while retaining the true hits. Most sequences are rejected by the filters and *filtration efficiency* is measured by the fraction of retained sequences in the filtered database. The 20-year history of database search in genomics is essentially the history of designing more and more efficient and statistically sound filters. From this limited perspective, the history of MS/MS database search is at the very beginning. Good filters for MS/MS database search are not trivial, and

until recently the studies of peptide identification have concentrated primarily on scoring [8, 47, 132, 162, 182, 191, 213, 236, 237], or the reliability of the peptide assignments [118, 137, 149, 175], with little focus on filters.[1]

We argue that a study of filtration is central to PTM identification. At first glance, this is counter-intuitive since there is no apparent connection between reducing the number of candidates and identifying modified peptides. Note, however, that aggressive (but accurate) filtration allows us to apply more sophisticated and computationally intensive algorithms and scoring to the few remaining candidates. Indeed, the current approaches to PTM analysis are based on generating huge "virtual databases" of all PTM variants. As Yates and colleagues remarked in [237], extending this approach to a larger set of modifications remains an open problem. However, if the database is reduced to a few peptides, one can afford to consider all possible PTMs for every peptide in the filtered database.

Mass-spectrometrists routinely use *peptide sequence tags* (PSTs) for spectra interpretation. The idea of using Peptide Sequence Tags as filters is not novel (see for examples Mann and Wilm [139], Mørtz et al. [147], Clauser et al. [33]), and has been studied recently in different forms [50, 96, 192, 208, 212]. In particular, Tabb et al. [212] recently released the GutenTag algorithm for PST generation and raised the possibility that searching with PSTs as filters results in additional identifications [212]. However, while these new tools greatly improve on early heuristics for PST generation, they do not explicitly measure the filtration efficiency versus accuracy trade-off, and the final analysis produces fewer hits than Sequest, albeit with fewer false positives. We emphasize that not every set of PSTs considered in refs. [192, 208, 212] forms a filter since it does not necessarily satisfy the *covering* property that ensures with high probability that the correct peptide is not filtered out. As a result these tools cannot eliminate a need to run a time-consuming database search (like Sequest) but rather provide *additional candidates* that Sequest may miss. Our goal is to completely substitute Sequest with a few orders of magnitude faster filtration algorithm. Below we describe the first steps toward this goal. In particular, our tag generation algorithms significantly improve on GutenTag [212] and

[1] At best, the use of parent peptide mass and trypsin end-point specificity could be thought of as simple filters in existing database search engines.

lead to efficient database filtration.

Our new MS/MS filtration tool combines the following components (i) de novo peptide sequencing, (ii) accurate PST generation, (iii) trie-based database search with PSTs, (iv) extension of tags to generate peptide candidates, including modified ones, and (v) scoring of the PTM candidates in the filtered database.

3.B Methods

3.B.1 Covering Set of Tags

While PepNovo improves on the existing de novo algorithms, using de novo predictions as text-based filters is hardly possible since they often contain errors caused by limitations of scoring functions. A single sequencing error can render de novo results useless in the text-based filtration mode for a database searches. However, it is not necessary to reconstruct the entire peptide sequence that created a spectrum. The filtration can be done equally well with a partial sequence if it is known to be correct with high probability. Below we formulate the problem of constructing such high probability PSTs that can serve as filters.

The paths in the properly constructed spectrum graph represent all possible peptides. While de novo algorithms find the best-scoring path amongst all these paths (de novo sequencing), the path corresponding to the true peptide is not necessarily the highest scoring one. It is not clear how to find the best-scoring path among paths corresponding to the database peptides (peptide identification) without testing each such "database path" in a case-by-case fashion. The tag generation offers an alternative that alleviates this time consuming step.

If the score of the best-scoring "database path" is δ, let $\mathcal{P} = \mathcal{P}(\delta)$ be a set of all paths in the spectrum graph whose score is larger than or equal to δ. We call this set δ-*suboptimal* and emphasize that while δ is not known in advance, one can compute a lower bound for δ. The question then arises whether there exists a simple characterization of the set $\mathcal{P}$ that would allow us to efficiently filter the database and to retain only database sequences from $\mathcal{P}$? For example, if all paths in $\mathcal{P}$ contain a tri-peptide SEQ then one can safely filter the database retaining only the peptides with

SEQ at a certain prefix mass and score these few remaining peptides. In reality, one PST may not be sufficient and we are interested in a *covering set* of PSTs X such that each path from $\mathcal{P}$ contains at least one PST from X. In this paper we propose a probabilistic solution to the problem of tag generation. The question arises on how to identify the most reliable amino acids in the de novo reconstruction and use them for tag generation.

3.B.2 Reliability of Amino Acids in De Novo Predictions

While de novo algorithm return the highest scoring path in the spectrum graph, in many cases it may be an incorrect reconstruction. However, for the purpose of PST generation, we do not need the whole prediction to be correct, rather just a sufficiently long portion of it. In such cases it is important to be able identify the most reliable amino acids in the reconstruction.

Every predicted amino acid corresponds to an edge in the spectrum graph (See Section 2.A.2 for more details about spectrum graphs). One way to determine if an edge is correct is by assessing how important it is for obtaining a high scoring path in the spectrum graph. If we remove a correct edge from the spectrum graph, the true peptide's path no longer exists. A subsequent run of the de novo algorithm on the modified graph should yield a de novo reconstruction with a significantly lower score (since the correct path is now disrupted, and all that is left are random spurious paths). However, if the edge is incorrect, its removal should not cause a large score reduction since there should exist a relatively good alternative (the correct path). It turns out that the ratio of the reduction between these two scores (called **Score Reduction due to Edge Removal**) correlates well with the reliability of predicted amino acids. The Score Reduction due to Edge Removal is not the only *feature* correlated with reliability of amino acids and below we describe some other features. The transformation of these features into a probabilistic estimate is not a trivial problem. We use the logistic regression method [92] to transform the combination of feature values into probabilities.

In order to determine the reliability of amino acid assignments, we view this task as a classification problem. The training samples in this scenario are pairs of variables of the form (x, y), where x is an amino acid from an input space $\mathcal{X}$, and y is a class variable from $\mathcal{Y} = \{0, 1\}$, which states if the sample is correct ($y = 1$) or incorrect ($y = 0$). The

reliability assessment task is reduced to creating a probabilistic model that determines for an unknown sample $x \in \mathcal{X}$ the probability $p(y = 1|x)$, which is the probability that x is correct amino acid. To use the logistic regression, we map each sample x into a point $\bar{x} \in \Re^n$, using n *feature functions* of the form $f : \mathcal{X} \rightarrow \Re$. The probability function derived from the regression model is

$$p_\lambda(y = 1|x) = \frac{e^{\lambda_0 + \sum_{i=1}^{n} \lambda_i \bar{x}_i}}{1 + e^{\lambda_0 + \sum_{i=1}^{n} \lambda_i \bar{x}_i}}. \tag{3.1}$$

The $\bar{x}_i$ are the feature values given to x by the n feature functions. The λ parameters in Eq. 3.1 are fit according to the training data using a nonlinear Conjugate Gradient method [196].

Logistic regression models maximize the training data's log-likelihood, given by $\sum_{(x,y)} \log p_\lambda(y|x)$ where the sum is over all training samples. The success of these models in assigning points to their correct classes depends a lot on the quality of the feature mapping, in particular on the features' ability to capture the nuances that distinguish between correct and incorrect predictions. Following is a description of features we use in our model.

Score Reduction due to Edge Removal. (described above).

Cleavage Site Scores. We observed that edges connecting two high scoring vertices in the spectrum graph are likely to be correct, while erroneous edges often connect high and low scoring vertices. If the lower score amongst the two is still relatively high, this is a good indicator that we have a correct amino acid (the two features we create are *high* and *low* PepNovo vertex scores).

Consecutive Fragment Ions. Correct interpretations of spectra often contain runs of b and y-ions. Incorrect interpretations have fewer such runs due to spurious cleavage sites. Therefore the detection of b-ions or y-ions at both ends of the edge is an indication of an accurate assignment. The bb feature is an indicator function equal to 1 iff the b-ions on both ends of the edge are detected. The yy feature is defined similarly.

Peak offsets. Since the peaks' positions in MS/MS spectra are imprecise, we use a tolerance of ± 0.5 Da in the peak location. However, the series of b or y-ions tend to have accurate offsets from each other (i.e., the mass difference between the consecutive ions is close to the mass of the amino acid). These offsets are usually larger for incorrect amino acid assignments. We define two features, the first is the squared offset of the b-ions, and the second is the squared offset of the y-ions.

Amino Acids Indicators. Some amino acids are more likely to be involved in incorrect de novo predictions than others. For instance, the amino acids glutamine and tryptophan have masses that are equal to the sum of two other amino acids. Some erroneous predictions with these amino acids involve cases where these amino acids were used instead of the combination of amino acids. We created indicator functions that equal 1 if the amino acid is one of the above. Another type of indicator can be used to identify amino acids involved in the proteolytic digestion. For instance if we know that the peptides were obtained through tryptic digestion, we can add an indicator to the model that is 1 if the amino acid is at the C-terminal and is either lysine or arginine (since tryptic peptides usually terminate with these amino acids).

The features described above capture different aspects of a genuine amino acid assignment. The cleavage site score features are very powerful in the sense that they capture much of the phenomena that characterize genuine cleavage sites (e.g., logical combinations of fragment ions, intensities etc.). The other features go beyond the scope of single cleavage sites. The Consecutive Fragment Ions feature and the Peaks Offset features capture phenomena that involve the relationship between the two cleavage sites that define an amino acid, while the Score Reduction feature goes further and describes how the single amino acid interacts with the larger spectrum graph.

Since most of the features have unbounded continuous values, it is hard to gauge their importance from their weights λ assigned by the logistic regression. However, we can get an idea of their contribution by examining the results when each feature is removed. According to this measure, the most important features are the Cleavage site scores. Following them in decreasing order of importance are the score reduction feature, the amino acid indicators for the cleavage amino acids, the consecutive fragments

indicators, the squared offset features, and the least important were the remaining amino acids indicators.

3.B.3 Tag Generation

A peptide sequence tag (PST) is a short amino acid sequence with a prefix mass value designating its starting position in the whole peptide. When used for filtration, PSTs can be extremely efficient in reducing the number of candidate database sequences that need to be scored (see Section 3.C). In this section we describe how to generate these sequence tags and asses their reliability.

De novo peptide sequencing and PST generation are related but distinct problems. Although PST generation appears to be a special case of de novo sequencing, most PST generation algorithms do not use probabilistic modeling for PST generation. Recent approaches [192, 212] described some heuristics for tag generation, however they do not take advantage of recent advances in de novo sequencing.

In de novo sequencing, the goal is to select a single path, as long and accurate as possible. For filtering, we are interested in a small set of shorter *local* paths (tags) that satisfy the *covering* property: at least one tag in the collection is correct (so the true peptide will not be filtered out). Of course, one is interested in a small covering set of tags (otherwise the filtration is inefficient). Generation of small covering sets of tags is a tricky problem and the recent approaches to tag generation [192, 208, 212] did not explicitly address the covering condition. We argue that PST generation may greatly benefit from algorithms developed specifically for de novo sequencing.

Starting from Mann and Wilm, 1994 [139], all approaches to PST generation search for *local* tags without checking whether the tags are part of a *global* de novo interpretation. Our results below indicate that this approach may have inherent limitations since the tag generation algorithms based on global rather than local approach seem to perform better. An optimal local path may not be extensible into an optimal global path, or indeed into any global path at all. We call such a misleading local path a *garden path* (referring to 18^{th} century English maze gardens with many dead-end paths.)

Similarly to our method for estimating the reliability of amino acids (see previous section), we would like to estimate reliability of the generated tags. The approach

that simply multiplies the probabilities of the individual amino acids in the tag does not produce good results, since it assumes that the amino acids are independent. In many instances this is not the case (e.g., in a tag SEQ, the amino acid E shares a cleavage site both with S and with Q). We take a different approach based on the adage *"A chain is only as strong as its weakest link"* because all it takes to render a tag incorrect is for one of its amino acids to be wrong.

The features we use in the tag generation model are as follows: (i) The lowest probability amongst the amino acids in the tag (the weakest link); (ii) The probability of the neighbor of the weakest link (if it has two neighbors we choose the neighbor with the lowest probability amongst the two); (iii) The geometric mean of the probabilities of the remaining amino acids in the tag. Using these features we train a logistic regression model to evaluate the reliability of a tag.

We explored three different approaches for tag generation. The first one (called PepNovoTag) exploits the fact that PepNovo is quite accurate in its predictions (in our test data, 72.7% of the amino acids are correct and 53.9% of all substrings of length 3 are correct). Therefore, it is likely that tags derived from PepNovoTag will also be correct. PepNovoTag extracts all substrings of the desired length from the PepNovo reconstruction and assigns probabilities to these tags using the logistic regression model. Because its PSTs are taken from a de novo reconstruction, PepNovoTag is not misled by garden paths.

In the second method (called LocalTag), the vertices of the spectrum graph are scored according to PepNovo's scoring. The spectrum graph is then searched, all sub-paths of the desired length are extracted as tags, and probabilities are assigned to the tags using the regression model. This tag generating method requires changes to the previously described probability models we use to asses the reliability of amino acids and tags. The "Score Reduction due to Edge Removal" feature in the amino acids model cannot be used since it requires the amino acid in question to be part of a complete high scoring path. Another change we made was to add a feature to the tag probability models. In PepNovoTag, because the tags are derived from a de novo path, they almost always have the correct orientation. However, when we extract tags from

the spectrum graph in LocalTag, it is likely that both a correct tag and its mirror[2] have a high score and we cannot really tell which one is correct. Usually the tag with the correct orientation has a slightly higher score (it is typical to detect stronger y-ions than b-ions, and PepNovo's scoring accounts for this). Therefore, we added to the LocalTag method a feature which measures the ratio of scores between the tag and its mirror (if it exists).

The third tag generating method, LocalTag+, merges PepNovoTag and LocalTag lists of tags into a combined list of tags that is sorted according to the tags' probabilities.

3.B.4 Database Filteration

In combinatorial pattern matching, matching a thousand patterns against a database takes roughly the same time as matching a single pattern. This speedup can be achieved using Aho-Corasick [3] algorithm that preprocesses the set of patterns to construct a *trie*. We construct a trie of all PSTs in multiple spectra and use it to search the protein database for all the spectra's tags simultaneously. While scan time does not increase with a larger number of tags, the number of peptide candidates increases, which in turn increases the scoring time. Therefore, we also employ a *tag extension* step, analogous to *seed extension* in sequence similarity search. The sequence tag has the prefix mass, and a scan can tell us if the prefix substrings have the right mass. This is trickier in the presence of PTMs [169], and we use dynamic programming to scan efficiently. Further details regarding the database scanning, tag extension, and practical applications of this approach to search for PTMs in large samples of MS/MS spectra are given in Tanner et al. [219], which describes the implementation of the InsPecT database search algorithm.

[2]The mirror tag is the tag obtained when the roles of the b and y-ions are reversed.

3.C Experimental Results

3.C.1 Data Set and Model Training

Our main dataset is the same one for PepNovo's benchmark experiments (Section 2.C). It consists of doubly charged tryptic peptides from ISB dataset [119] and the Open Proteomics Database (OPD) [171]. In total we obtained 1252 spectra of peptides with unique sequences which were identified by Sequest with $X_{\text{corr}} > 2.5$. From this set 280 spectra were set aside as the test set. In addition, we also used a recently released protein mixture dataset [173] as an independent test set for some of our experiments.

The regression models we used both for amino acids and tags were trained using the 972 spectra of the training set. We ran our de novo algorithm PepNovo on each of the spectra, and positive samples were created for each of the correctly predicted amino acids (an amino acid was considered correct if its location in the predicted sequence was within 2.5 Da from its location in the true peptide that created the spectrum). Likewise, negative samples were created for the amino acids that were predicted incorrectly. A nonlinear Conjugate Gradient method [196] was used to train the parameters for the regression model for the amino acids.

Two separate sets of samples were created to train the PepNovoTag and LocalTag models. PepNovoTag samples were created by parsing PepNovo's results on the spectra in the training set to create tags. The LocalTag samples were created by selecting from each spectrum's spectrum graph the highest scoring local tags (the number of tags selected was the same as the number of PepNovoTag tags generated for that spectrum). These two sets of samples were then used to train two regression models for the tag generating algorithms.

3.C.2 Reliability of Individual Amino Acids

We conducted the following experiment to asses the quality of our amino acid probability assignments. For each spectrum in the data set, we obtained a de novo prediction using PepNovo. The amino acids in the training (test) set were sorted according to decreasing predicted accuracy and divided into bins containing 200 amino acids each. Each point in Figure 3.1 represents a bin, its x coordinate is the average predicted

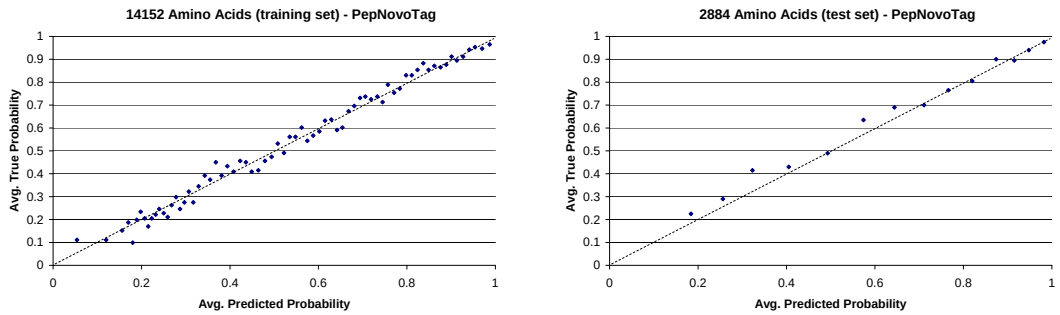


Figure 3.1: Comparison of the average predicted probability and the true probability of amino acids. Results are reported for the training set (*left*) and the test set (*right*).

probability that the samples in the bin are correct amino acids (calculated using the regression models), and the y coordinate is the true proportion of samples in the bin that are correct amino acids. The diagonal dash line represents the region where the predicted probabilities equal the true probabilities. In an ideal figure, obtained using an oracle for probability assignments, we would find two dense clusters in the graphs. The first, located near (0,0), would contain the incorrect amino acids, and the other, located near (1,1), would contain the correct amino acid assignments. However, in many cases it is difficult for our model to be that discriminating, and when confronted with questionable amino acids it resorts to assigning them probabilities throughout the [0,1] range.

3.C.3 Reliability of Tags

Figure 3.2 compares PepNovoTag with LocalTag (for tags of length 3). Two separate sets of tags were generated as follows. For each of the 280 spectra in the test set, PepNovo-generated tags were placed in the PepNovoTag tag list. In addition, an equal number of highest probability tags was extracted from the spectrum graph, and placed in the LocalTag tag list. Note that the composition of tags is not the same in both sets. Only 32.8% of the tags predicted by LocalTag are correct, compared to 53.9% correct tags predicted by PepNovoTag. In addition, the PepNovoTag probability model is much more robust than the LocalTag model. The mean probability assigned to a correct tag by PepNovoTag's model is 0.722 (with 30.2% of the correct tags given probability greater than 0.9), whereas the mean probability assigned to the correct tags in the LocalTag is

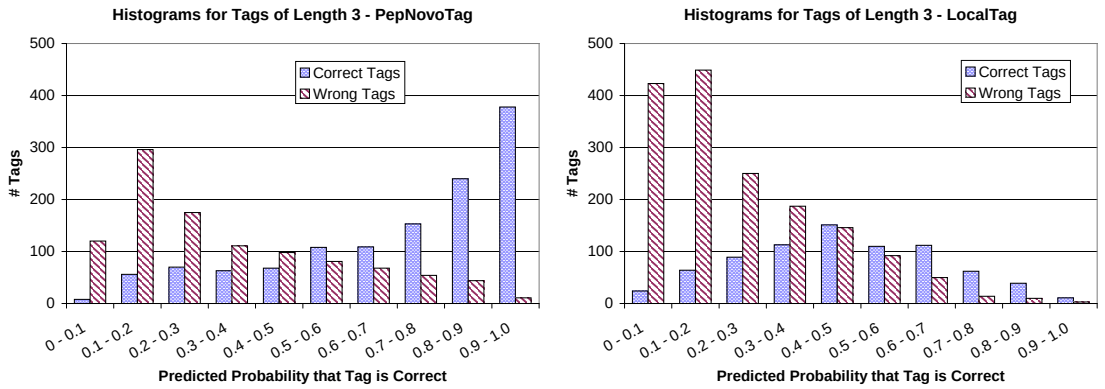


Figure 3.2: Histograms of predicted tag probabilities. Both histograms show the probabilities for 2307 tags of length 3 derived from the 280 spectra in the training set. On the left is the histogram of PepNovoTag tags (53.9% of these tags are correct), and on the right is the histogram of the LocalTag tags (32.8% of these tags are correct).

0.473 (with only 1% of the correct tags being assigned probabilities above 0.9). It is apparent that the PepNovoTag has an advantage over the LocalTag, since by knowing that the tag came from a de novo prediction, there is *a priori* higher probability that the tag is correct.

Figure 3.3 contains plots of the true probabilities of tags vs. the predicted probabilities by the regression models. Similarly to Figure 3.1, the tags were sorted according to the predicted probability, and were binned in bins of 200 tags. Each point in the plot represents such a bin, where the x coordinate is the average predicted probability, and the y coordinate is the proportion of correct tags in the bin. The plots on the top of the figure contain the data for the tags of length 3 that were used in Figure 3.2. The plots on the bottom of Figure 3.3 contain data for tags of length 6 where an equal number of PepNovoTag and LocalTag tags were generated from each spectrum in the training set (a total of 1467 tags). 31% of the tags generated by PepNovoTag were correct, compared to only 19% correct tags that were generated by LocalTag. For both tags lengths and both tag generating algorithms we see that our models are not biased since all points lie close to the diagonal. In addition the plots show that it is more difficult to make strong predictions of correctness for tags than it is to make them for amino acids (the points in this plot are farther away from the (1,1) corner). Since any single amino acid can render a tag incorrect, the models ability to assign high probabilities to correct tags diminishes

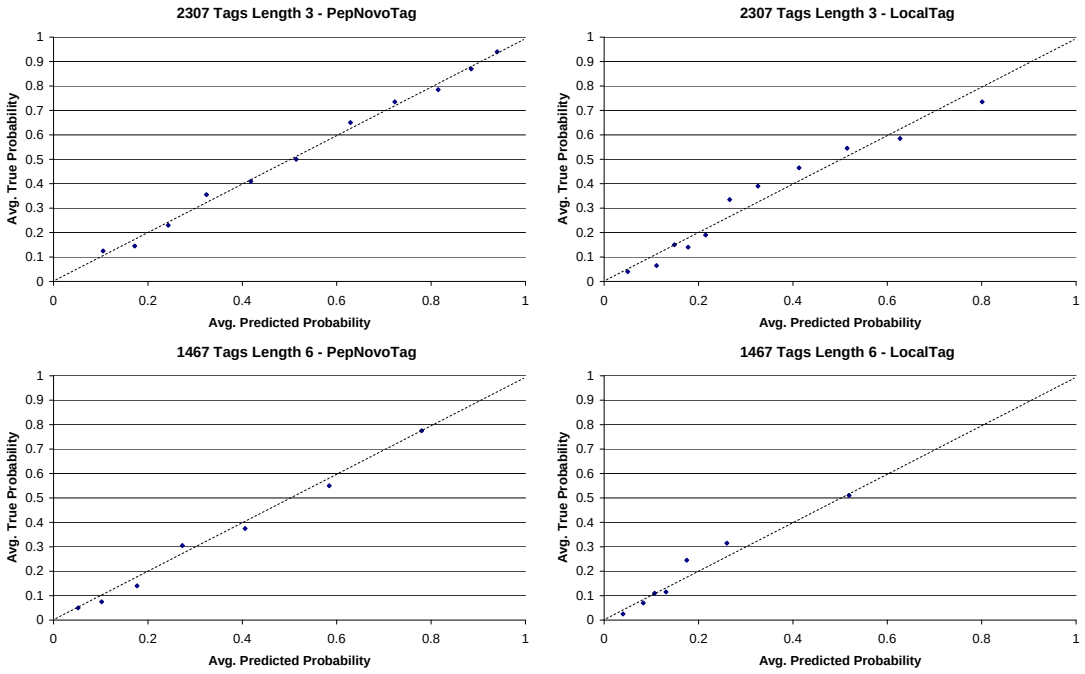


Figure 3.3: Comparison of the average predicted probability and the true probabilities of tags of length 3 (*top*) and length 6 (*bottom*). Results are reported for PepNovoTag (*left*) and LocalTag (*right*).

as the tags grow longer (see plots for tags of length 6).

3.C.4 Benchmarking Tag Generation Algorithms

In our probabilistic version of the Tag Generation problem, we would like to find a set of tags that are likely to cover the high scoring database paths, and in particular we would like the true peptide's path to be covered by at least one of our generated tags. It therefore makes sense to evaluate the merits of a set of tags based on the presence of at least one tag from the correct peptide.

Table 3.1 compares the performance of PepNovoTag, LocalTag, LocalTag+, and GutenTag [212]. PepNovoTag outperforms other methods when the number of tags that are generated is small. The highest scoring tag of length 3 generated by PepNovoTag is correct in 80.4% of the cases, compared to only 49.3% with GutenTag's highest scoring tag. Using 5 tags of length 3 generated by PepNovoTag, we obtain in 93.9% of the cases at least one correct tag compared to only 81.1% of the cases for GutenTag. Table 3.1

Table 3.1: Comparison of tag generating methods (280 spectra in the test sample). For each tag length, algorithm and number of generated tags, the table displays the proportion of test spectra with least one correct tag. Since the number of tags that can be generated by PepNovoTag is limited by the length of the predicted sequence, usually no more than 10 tags were predicted. GutenTag was run with the default settings for ion trap tryptic peptides. Due to the long time required to generate tags of length 6, this data was not collected for GutenTag.

Tag Length	**Algorithm**	**Number of Generated Tags**						
		1	3	5	10	25	50	100
3	LocalTag	0.529	0.764	0.929	0.957	0.971	0.975	0.979
	PepNovoTag	**0.804**	**0.925**	0.932	0.946	-	-	-
	LocalTag+	0.725	0.855	**0.939**	**0.961**	**0.979**	**0.979**	**0.982**
	GutenTag	0.493	0.732	0.811	0.893	0.914	0.936	0.950
4	LocalTag	0.464	0.714	0.771	0.850	0.932	0.943	0.954
	PepNovoTag	**0.732**	**0.850**	0.864	0.871	-	-	-
	LocalTag+	0.700	0.811	**0.871**	**0.900**	**0.946**	**0.954**	**0.964**
	GutenTag	0.418	0.614	0.711	0.782	0.832	0.861	0.879
5	LocalTag	0.410	0.593	0.678	0.786	0.836	0.854	0.879
	PepNovoTag	**0.664**	**0.764**	**0.775**	0.800	-	-	-
	LocalTag+	0.571	0.696	0.736	**0.803**	**0.846**	**0.864**	**0.893**
	GutenTag	0.318	0.464	0.539	0.643	0.736	0.761	0.775
6	LocalTag	0.332	0.489	0.593	0.661	0.739	0.771	**0.804**
	PepNovoTag	**0.579**	**0.632**	**0.639**	0.654	-	-	-
	LocalTag+	0.527	0.546	0.593	**0.671**	**0.743**	**0.779**	**0.804**
	GutenTag	-	-	-	-	-	-	-

suggests that if the desired number of tags is small (typically less than 5), PepNovoTag should be used. However, if a larger number of tags is desired, one should switch to the LocalTag+ which performs better with larger sets of tags.

Since interpreting mass spectra is a high-throughput process, it is worthwhile to discuss the running time required to generate the tags. Typically it takes PepNovoTag or LocalTag+ less than 0.1 seconds to generate a set of tags. LocalTag+ running time scales well with increasing tag lengths, where generating tags of length 6 takes less than 0.2 seconds. GutenTag on the other hand, doesn't scale well with increasing tag length. While it can generate tags of length 3 at a decent pace, it takes an average of 1 minute to generate tags of length 5, and in some cases more than 30 minutes to generate tags of length 6 for a single spectrum.

Due to the selection and validation criteria we used, the 280 spectra in our

Table 3.2: Benchmark experiments with an independent protein mixture dataset. A total of 685 mass spectra of doubly charged peptides with $X_{\text{corr}} > 2$ and $\Delta C_n > 0.1$ were taken from the new protein mixture dataset [173]. This set was broken down into three groups according to their Sequest X_{corr} score: (I) 121 spectra with low confidence ($X_{\text{corr}} < 2.5$), (II) 236 spectra of medium confidence ($2.5 \leq X_{\text{corr}} < 3.5$), and (III) 328 spectra of high confidence ($X_{\text{corr}} \geq 3.5$). For each tag length, algorithm, dataset and number of generated tags, the table displays the proportion of test spectra with least one correct tag.

Algorithm , Tag Length	**Test Set**	**Number of Generated Tags**						
		1	3	5	10	25	50	100
GutenTag , 3	I	0.174	0.289	0.347	0.430	0.537	0.579	0.603
	II	0.203	0.331	0.398	0.492	0.619	0.661	0.691
	III	0.409	0.595	0.646	0.704	0.756	0.799	0.832
	all 685	0.296	0.450	0.508	0.582	0.670	0.712	0.743
LocalTag+ , 3	I	0.595	0.661	0.702	0.793	0.843	0.843	0.860
	II	0.708	0.814	0.826	0.881	0.924	0.958	0.970
	III	0.841	0.899	0.927	0.939	0.960	0.973	0.988
	all 685	0.752	0.828	0.853	0.893	0.927	0.945	0.959
LocalTag+ , 4	I	0.488	0.587	0.653	0.694	0.793	0.810	0.810
	II	0.661	0.746	0.771	0.814	0.864	0.907	0.941
	III	0.756	0.860	0.884	0.921	0.945	0.957	0.966
	all 685	0.676	0.772	0.804	0.844	0.891	0.914	0.930
LocalTag+ , 5	I	0.388	0.430	0.479	0.570	0.661	0.702	0.711
	II	0.513	0.623	0.665	0.758	0.835	0.860	0.877
	III	0.695	0.793	0.820	0.878	0.918	0.930	0.942
	all 685	0.578	0.670	0.707	0.782	0.844	0.866	0.879
LocalTag+ , 6	I	0.298	0.347	0.413	0.504	0.554	0.570	0.620
	II	0.470	0.576	0.631	0.682	0.758	0.797	0.814
	III	0.601	0.716	0.765	0.835	0.887	0.899	0.915
	all 685	0.502	0.603	0.657	0.724	0.784	0.806	0.828

test might not represent the typical spectra in a real mass spectrometry experiment. In addition these spectra come from the same source as the training data, and thus might not give an accurate indication of our algorithm's performance on general data. For this reason we conducted additional benchmark experiments using spectra from the recently released protein mixture dataset [173]. This dataset contains spectra from a mixture of known peptides and proteins, with a consistence and an abundance that is typical of real mass spectrometry experiments. The spectra in this dataset where created by a different type of mass spectrometer than the previous ISB dataset, and these spectra also have different characteristics (they tend to have more peaks than the training spectra).

Table 3.2 contains the results of additional benchmarks we ran on spectra from the new protein mixture dataset. A total of 685 spectra of doubly charged tryptic peptides were extracted, the identifications to these spectra were given by running SEQUEST on a 1.5Mb sequence file. Since the dataset contains spectra of different quality and the identifications were done with different levels of confidence, we also partitioned the data according to the SEQUEST X_{corr} score (121 spectra with $X_{\text{corr}} < 2.5$, 236 spectra with $2.5 \leq X_{\text{corr}} < 3.5$, and 236 spectra with $X_{\text{corr}} \geq 3.5$.) The results in the table show that despite the different type of data, our algorithm still performs well. Though there is some drop in the performance compared to the results in Table 3.1, generally there is still a high success rate, with the performance gap narrowing as more tags are considered. When we examine the results for the different subsets of the dataset, we see that there is a large disparity between the results for the low confidence spectra (group I) and the high confidence spectra (group III). The results for group III are very similar to the success rates in Table 3.1, while the results for group I are much lower (they generally have a success rate that is 15-20% lower). Note that part of the deterioration can be attributed to the lower quality spectra in group I, compared to the other spectra that have a stronger signal. In addition it is likely that some of the spectra in group I are false positives (since they have a low X_{corr}), while this is less likely to occur with the other groups of spectra.

We also included benchmark results for GutenTag on this data (on tags of length 3). The increased number of peaks in these spectra proved detrimental to GutenTag's performance. This led to both a larger running times and considerably lower success rates compared to the results on the previous test set.

3.C.5 Database Filtration Results

The main purpose of our tag filtration method is to reduce the running time of database search algorithms. The following benchmark experiments are intended to measure this speedup.

In a typical filtration scenario, a single tag of length 3 used in PTM detection mode had on average 30000 hits to the 54Mb SWISS-PROT database (without using prefix mass values as filters). Because we consider peptides of various lengths including

Table 3.3: Efficiency of tag-based filtration. Spectra from the ISB data set were searched against the SWISS-PROT database (54 Mb) using a standard desktop PC (3GHz CPU). The search permitted one or both endpoints to be non-tryptic, and allowed missed cleavages; requiring tryptic endpoints would further improve filtration efficiency. The database filtration was done as a batch job using a single scan of the database to find the occurrences of the tags from all the query spectra. The reported runtime is the average time required to perform the filtration for a single spectrum.

PTMs	Tag Length	# Tags	# Candidates	Filtration Efficiency	Runtime
None	3	1	181	3.4×10^{-7}	0.17s
	3	10	888	1.6×10^{-6}	0.27s
	4	1	10	1.9×10^{-8}	0.26s
	4	10	60	1.1×10^{-7}	0.89s
Phosphorylation	3	1	311	5.8×10^{-7}	0.21s
	3	10	1480	2.7×10^{-6}	0.38s
	3	25	2650	4.9×10^{-6}	0.60s

non-tryptic peptides, the effective database size is roughly 550 million entries. Such a large number of entries requires efficient filtration in order to obtain results in reasonable time. Table 3.3 gives examples of the efficiency of our tag filtration, along with the running time required to perform this filtration. For example, using a single tag of length 3 as a filter yields on average 181 candidate peptides having both a correct parent mass and a correct prefix mass for the tag. Of course, a single tag often does not satisfy the covering condition, particularly for low-quality spectra. Increasing the number of generated tags to 10 ensures with high probability that the resulting set satisfies the covering condition and still provides high filtration efficiency of 1.6×10^{-6}. This is almost two thousand-fold more efficient than using only the parent mass as a filter (which has 0.003 filtration efficiency).

Considering post-translational modifications does not impact the number of initial matches, but affects the chances of a successful extension (and hence the scoring time). As annotated spectra of modified peptides are not readily available, we report statistics from a simulated dataset with phosphorylations introduced to the ISB spectra in a realistic probabilistic setting. Tag-based filters provide far greater efficiency in the presence of PTMs. For a case of up to two phosphorylations, 10 PSTs of length 3 are 1500 times as efficient as basic parent mass filtering. Each possible modification enriches the spectrum graph with more edges. For instance, phosphorylation adds three new

masses to our "alphabet" of possible edge masses (we considered phosphorylations of Serine, Threonine, and Tyrosine). Therefore, some increase in number of tags generated is necessary in order to maintain the same high sensitivity for medium-quality spectra. Twenty-five tags on the phosphorylated data set produce accuracy equivalent to ten tags on the unmodified data set (data not shown).

Although this test is run on simulated spectra, to the best of our knowledge, it is the first systematic benchmarking for speed and sensitivity of PTM identifications. Previous studies report identification of PTMs, but not how many PTMs are missed in the analysis. Thus, the sensitivity of these algorithms remains unknown. Tanner et al. [219] supplement our study through analysis of real PTMs in large data sets of spectra of modified peptides.

3.D Discussion

Our algorithm solves a probabilistic version of the Tag Generation Problem. Rather than search directly for a covering set of tags we attempt to find a set of tags that have a high probability of being correct (and are therefore likely to cover the high scoring database paths and in particular the path corresponding to the correct peptide). Our results show that our algorithm is quite successful at this task. According to Table 3.1 using just 10 tags of length 3, we cover 96.1% of the peptides. The results in Section 3.C.5 show that using 10 tags for filtration we can reduce the number of candidate peptides we need to score by a factor of almost two thousand. This is a huge saving in computations (since the scoring of peptide-spectrum matches is often costly), and it is achieved at the slight cost of loosing less than 4% of the positive identifications. Our benchmarking experiments also show that our tag generating methods are both more accurate and much faster than the GutenTag tag generation algorithm.

The filtration process we propose is aimed primarily at increasing the speed of database searches, so as to make them practical in scenarios where the search space is large (for instance, when considering post-translational modifications). Despite the filtration rate, our tagging can still be insufficient for searching very large sequence databases like a six-frame translation of the human genome (even with tagging, such

a search can take several minutes per spectrum). To increase the efficiency, we have to increase the lengths of the predicted tags. In the subsequent chapters we explore to methods in which this can be done. In Chapter 4 we examine de novo sequence with high-precision data which enables us to create longer sequence tags (6-8 amino acids), without compromising the accuracy. In Chapter 6 we see how advanced scoring models can enable us to increase the performance of de novo sequencing and tag generation to a degree that allows us to predict longer tags, even with low-resolution MS/MS data.

This chapter, in full, was published as "Peptide Sequence Tags for Fast Database Search in Mass-Spectrometry". A. Frank, S. Tanner, V. Bafna, and P. Pevzner. *Journal of Proteome Research*, 4:1287-95, 2005. The dissertation author was the primary author of this paper.

4

De Novo Sequencing With Precision Mass Spectrometry

4.A Introduction

In the last decade, tandem mass spectrometry (MS/MS) has emerged as a technology of choice for high-throughput proteomics. The precision and resolution of mass spectrometers are key parameters that draw a line between what is possible and what is impossible in MS/MS-based proteomics today. Instruments like the Quadruple Time-of-flight (QTOF) mass spectrometers are capable of accuracy in the range of a few parts-per-million [201]. Continuous efforts to improve mass resolution recently resulted in the breakthrough development of Fourier transform MS techniques, including magnet-based ion cyclotron resonance (ICR) instruments [140] and electrostatic FT traps (Orbitraps) [154], that improve resolution by two to three orders of magnitude as compared to conventional mass spectrometers. Emergence of *precision mass spectrometry* heralds a new era in proteomics and makes it possible to address the problems that were previously beyond the reach of traditional MS techniques.

Traditionally, there have been two major approaches to peptide interpretation: the database search and de novo sequencing. However, this separation is somewhat artificial since the de novo search can be viewed as a search in the very large database of all possible peptides. In recent years the boundaries between these two methods

have started to blur with de novo sequencing being used to generate tags for database filtration [33, 75, 139, 147, 212] and for homology based BLAST like searches [89, 192, 194, 208]. However, using de novo sequencing directly for peptide identification is not widely practiced; low resolution, incomplete fragmentation, and homeometric peptides (which we define below) make de novo approaches less accurate than the database search. Currently even the leading de novo algorithms correctly call only 70-75% of the amino acids (see Chapter 2).

With its significantly higher accuracy and resolution, precision mass spectrometry offers the opportunity for superior sequencing performance. However, since precision mass spectrometry is a relatively new area, there is still a shortage of publicly available FT-ICR and OrbiTrap datasets and computational tools geared toward these new instruments. Moreover, accurate de novo sequencing with precison mass-spectrometry remains a challenge. Indeed, previous sequencing approaches for precision mass spectrometry data required particular experimental setups, such as the use of dual fragmentation pathways (CAD/ECD) for de novo sequencing [99, 185]. Other approaches are based on computing amino acid composition [155, 202], thus making them accurate, but rather slow for high-throughput sequencing. Also, these algorithms did not take advantage of the spectrum graphs, the key computational technique behind de novo peptide sequencing. In this chapter we apply the powerful spectrum graph techniques to precision mass spectrometry and argue that precision mass spectrometry calls for development of new computational ideas for peptide identification. In particular, we show that the percentage of error-free peptide identifications increases from approximately 30% for traditional MS instruments to 90% for precision mass-spectrometry. Recently, Savitski et al. [185, 186], proposed a de novo algorithm for a special experimental setup for FT based on complementary fragmentation methods (ECD and CAD). In this work they were able to overcome the problems associated with the incomplete fragmentation of stand-alone CAD and produce accurate peptide reconstructions. Our approach achieves similar accuracy in the standard spectral acquisition mode that is well amenable to high-throughput analysis.

With the current methodology, an MS/MS database search compares every mass spectrum against every peptide in a database (within a specified precursor mass

tolerance) so the running time typically scales linearly with the database size and exponentially with the number of post translational modifications (PTMs) considered. This makes comparison of millions of spectra against many peptides computationally prohibitive. Recently developed MS/MS database search tools such as X! Tandem [43, 44] and InsPecT [219] achieve orders of magnitude reduction in the running time of peptide identification by using filtration methods. Using precision mass spectrometry can greatly reduce the computational cost of database searches by taking advantage of the accurate precursor mass measurements to eliminate a larger proportion of the database peptides from consideration.

In this work we explore a different approach to database search, which delivers fast and accurate peptide identification. Our algorithm capitalizes on precision mass spectrometry to generate accurate de novo sequences for each query mass spectrum. These sequences are compared to the database using fast pattern matching (e.g., hash table lookup), as opposed to slow spectra matching. The bulk of our algorithm's analysis is performed by de novo sequencing (that is very fast), so the running time is practically independent of the database size. The difference between the traditional approach and our de novo based approach is illustrated in Figure 4.1.

Having running time independent of the database size is an important advantage over the traditional MS/MS database search algorithms. However, this advantage is less crucial for traditional database searches with precision MS/MS since the accurate precursor mass serves as a filter to reduce the number of explored variants. More important is an ability to analyze peptides that are *not* in the database, e.g., alternatively spliced variants, fusion proteins, programmed frame shifts, etc. While traditional database search often fails in such cases (the effective database size in such applications is too high to be explicitly generated), our approach opens a possibility to address them with combinatorial pattern matching algorithms. For example, Tanner et al., 2007 [218] recently succeeded in identifying new alternatively spliced genes via MS/MS analysis. However, the database in this case includes all putative (potentially overlapping) exons in human genome and all putative splice junctions. With our approach, the search for alternative splicing can be reduced to a simple version of the spliced alignment problem, a well studied problem in genomics.

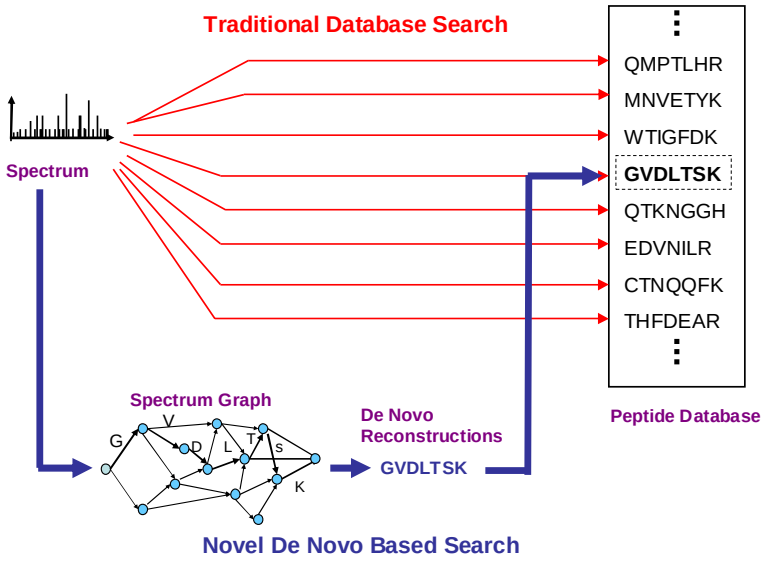


Figure 4.1: Two approaches to peptide identification. The traditional approach based on comparing spectra to the database (red) vs. our approach based on de novo sequencing and fast database lookup (blue).

4.B Methods

4.B.1 Homeometric Peptides

We first introduce the concept of *homeometric peptides* that are different peptides with similar theoretical MS/MS spectra, which can induce sequencing errors both with de novo and database search algorithms. We show that homeometric peptides are abundant making it inherently impossible to design an accurate de novo sequencing algorithm that outputs a single peptide as a solution. We therefore argue that peptide sequencing algorithms should output multiple solutions and show how to design such algorithms.

For a peptide P of length k, let $\overrightarrow{P} = \{P_1, P_2, \ldots, P_k\}$ be the set of all prefix masses of P, and let $\overleftarrow{P}^{18} = \{P_{-1} + 18, P_{-2} + 18, \ldots, P_{-k} + 18\}$ be the set of all suffix masses of P plus a mass of 18 Da[1]. Given a mass tolerance threshold ε and two sets of masses $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$, we say that $X \approx Y$, if $|x_i - y_i| < \varepsilon$ for

[1]The prefix masses correspond to the $N-$terminal $b-$ion series and the suffix masses correspond to the $C-$terminal $y-$ion series.

$1 \leq i \leq n$. We say that a set X does not explain a mass y if $|x - y| > \varepsilon$ for every $x \in X$. The distance between sets X and Y is defined as the number of elements in Y not explained by X plus the number of elements in X not explained by Y. Peptides P and Q are called *homeometric* if $\overrightarrow{P} \cup \overleftarrow{P}^{18} \approx \overrightarrow{Q} \cup \overleftarrow{Q}^{18}$, i.e., if P's and $Q's$ theoretical spectra are the same (up to a mass tolerance threshold ε). Peptides P and Q are called *δ-homeometric* if the distance between $\overrightarrow{P} \cup \overleftarrow{P}^{18}$ and $\overrightarrow{Q} \cup \overleftarrow{Q}^{18}$, is less than δ, i.e., P's and $Q's$ theoretical spectra are the same up to a mass tolerance threshold ε, except for δ mismatched peaks.

Homeometric peptides are ubiquitous in low precision settings. For instance, there is over a 30% chance that an arbitrary peptide of length 10 has a homeometric peptide (see Figure 4.3). These percentages grow if we loosen the requirements and consider δ-homeometric peptides for small δ. A simple way to generate δ-homeometric peptides (for $\delta = 2$) is to swap adjacent amino acids in the peptide. However, more subtle instances of homeometric peptides can be created by switching between prefix and suffix vertices in the *spectrum graph* (see Section 4.B.2 for a definition of spectrum graphs). Figure 4.2 (a) shows an illustration of a mass spectrum for the peptide DHGMPF, and part (b) depicts the spectrum graph created from the b- and y-ions of that peptide. The graph contains two paths, the path of prefix masses (blue), and the reverse path of suffix masses (red). However in addition to these paths, there exists a path DFMGSF representing a homeometric peptide that "mixes and matches" prefix and suffix paths. Figure 4.2 (c) shows a rearranged version of the spectrum graph that gives a better understanding how the path for the homeometric peptide is obtained: The path for the peptide DFMGSF starts at the prefix path, crosses over to the suffix path (using amino acid F), traverses the suffix path (amino acids MG), returns to prefix path (using amino acid S) and continues along the prefix path.

Figure 4.2 illustrates that the key for having homeometric peptides is a pair of crossover edges between the prefix and suffix paths' vertices (these crossover edges also lead to symmetric paths for which the antisymmetric peptide sequencing algorithms were developed [30].) As observed by Budnik et al., 2002 [24], the crossover edges are quite common, making the confident de novo sequencing of many peptides impossible.

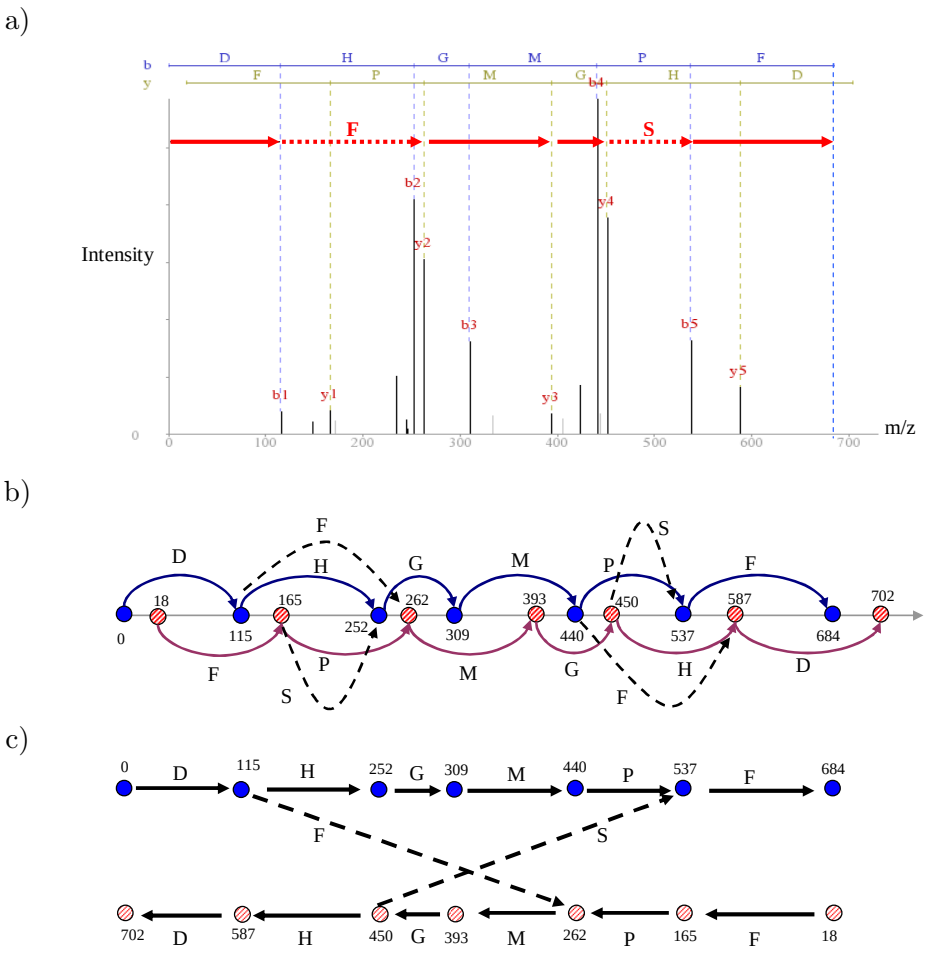


Figure 4.2: Homeometric peptides. a) Illustration of mass spectrum of DHGMPF, the red path shows the derivation of the peptide DFMGSF which starts with by using b_1, crossing over to the $y-$ ladder using y_2,y_3, and y_4, and then returning to the $b-$ions to b_5. b) The spectrum graph derived from the mass spectrum of DHGMPF. c) The same spectrum graph with vertices rearranged to show the relationship between the paths of the homeometric peptides DHGMPF and DFMGSF. The top path (0,115,252,...) represents prefix masses while the bottom path represents suffix masses of DHGMPF (the masses are rounded off to integer values).

4.B.2 De novo Peptide Sequencing With Precision Mass Spectrometry

De novo peptide sequencing is a fast alternative to the database search (although in most cases it produces less accurate results [73]). Most de novo algorithms model all possible peptides as paths in a *spectrum graph*, a directed acyclic graph with vertices corresponding to putative prefix masses (cleavage sites) of the peptide [13, 49].

Two vertices are connected by a directed edge from the vertex with the lower mass to the one with a higher mass if the difference between them equals the mass of an amino acid. Dancik et al., 1999 [49] describe in detail the construction and scoring of the spectrum graph. Since peptide fragmentation is often incomplete, the spectrum graph may be disconnected. For this reason we add edges corresponding to masses of pairs (triples, etc.) of amino acids. With the high resolution of FTMS we can use edges of up to three amino acids (which compensate for up to two consecutive missing backbone cleavages) without significant increase in computational complexity.

De novo algorithms attempt to find a peptide P that maximizes the probability of generating the query spectrum (under a certain probabilistic model). Dancik scoring [49] is based on a rigorous probabilistic model for computing this probability from fragment ion propensities defined in Table 4.2. The model we use implements two simple extensions to their basic scoring model. The first extension incorporates peak ranks into the scoring model. The second extension is to add the modeling of dependencies between fragments using the probabilistic model of the PepNovo algorithm [73].

Considering peak intensities improves scoring, since high intensity peaks are likely to represent y and b ion fragments. However, large variance in the absolute peak intensities exhibited in mass spectra makes it difficult to account for them in a framework of a rigorous probabilistic model. For this reason, peak intensities need to be normalized before being scored. From our experience, using the peaks' relative ranks in the spectrum, rather than their actual absolute intensities, gave optimal results in the scoring we used (compare to Tanner et al., 2005 [219]). We incorporated the peak ranks into the Dancik scoring using the distribution of peak ranks according to the fragment types as defined in Table 4.1 (see Section 4.C.2 for further details on the selection of these fragment types).

The Dancik scoring models different fragment ions as independent random variables. In practice, this assumption is often violated, for example, the variables corresponding to b- and y-ions are highly correlated. We used the probabilistic network structure of the PepNovo algorithm [73] to incorporate such fragment correlations into our scoring model.

Our de novo sequencing algorithm finds the highest scoring path in the spectrum

Table 4.1: Distribution of peak ranks according to fragment ions. Statistics were collected from 376 FT-ICR spectra of unique doubly charged peptides. We grouped peak ranks into a small set of 8 rank levels as follows: I) the peak ranked 1, II) ranks 2-3, III) ranks 4-7, IV) ranks 8-12, V) ranks 13-20, VI) ranks 21-30, VII) ranks 31-55, VIII) ranks 56-∞.

Ion	Peak Ranks							
	1	2-3	4-7	8-12	13-20	21-30	31 - 55	56 - ∞
y	0.838	0.702	0.365	0.187	0.100	0.060	0.040	0.010
b	0.066	0.102	0.265	0.300	0.181	0.098	0.066	0.020
$b - H_2O$	0.005	0.019	0.047	0.063	0.088	0.089	0.061	0.010
$y/2$	0.000	0.008	0.033	0.029	0.050	0.067	0.054	0.061
$y - H_2O$	0.003	0.007	0.021	0.030	0.043	0.045	0.035	0.040
y^{+2}	0.019	0.040	0.035	0.029	0.029	0.024	0.015	0.020
$b - NH_3$	0.000	0.003	0.013	0.024	0.027	0.026	0.029	0.020
a	0.000	0.004	0.007	0.020	0.023	0.020	0.023	0.030
$[y - H_2O]^{+2}$	0.003	0.004	0.015	0.016	0.014	0.016	0.014	0.030
$[y - H_2O - H_2O]^{+2}$	0.005	0.009	0.013	0.014	0.016	0.013	0.007	0.020
$b - H_2O - H_2O$	0.000	0.003	0.004	0.005	0.009	0.015	0.025	0.020
$y - NH_3$	0.000	0.003	0.003	0.010	0.014	0.009	0.015	0.010
$[y - H_2O]^{+2}$	0.000	0.003	0.005	0.011	0.009	0.005	0.004	0.000
$b/2$	0.000	0.000	0.001	0.004	0.004	0.008	0.015	0.020
$b - NH_3 - H_2O$	0.000	0.000	0.001	0.002	0.007	0.010	0.014	0.000
$a - NH_3$	0.000	0.000	0.003	0.002	0.004	0.006	0.007	0.010
$a - H_2O$	0.000	0.001	0.000	0.003	0.004	0.004	0.005	0.000
$[y - NH_3]^{+2}$	0.000	0.000	0.001	0.002	0.003	0.003	0.004	0.000
$b - NH_3 - NH_3$	0.000	0.001	0.000	0.002	0.001	0.003	0.003	0.000
b^{+2}	0.003	0.003	0.002	0.003	0.003	0.005	0.004	0.010
$y - H_2O - NH_3$	0.000	0.000	0.000	0.003	0.002	0.001	0.004	0.000
$y - H_2O - H_2O$	0.000	0.000	0.002	0.001	0.001	0.004	0.001	0.000
Unexplained	0.059	0.089	0.164	0.241	0.365	0.470	0.552	0.667
Total	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

graph in time linear in the number of edges[2]. Since the path may contain double and triple edges we define it as a correct reconstruction if all vertices in the path correspond to correct cleavages in the peptide. As a result our reconstruction represent amino acid sequences with gaps corresponding to masses of double and triple edges in the spectrum graph.

The gapped peptide P found by our algorithm is correct for 90% of spectra (see Table 4.3). However, in most of the remaining 10% of spectra the optimal path uses

[2]Note that we ignore the problem of symmetric paths since they are very rare with precision MS data (symmetric paths are formed when single peaks are used with multiple interpretations, e.g., a peak appears in one of the path's nodes as a $b-$ion and in another node as a $y-$ion). However, if solution paths are required to be anti-symmetric, the method of Chen et al., 2001 [30] can be used (it runs in time proportional to the product of the number of edges and the number of vertices in the spectrum graph.)

a single incorrect vertex, thus indicating that the optimal path usually comes close to the path representing the correct solution and represents a δ-homeometric peptides for a small δ. Since the difference in score between the optimal path and the correct solution is usually small, we advocate the search for suboptimal paths in the spectrum graph as potential peptide reconstructions. We empirically found a bound σ for the maximal score difference between the highest scoring optimal path and the correct suboptimal path (σ was set to the maximum score difference found in our training data). Using this threshold, we can remove all vertices from the spectrum graph that do not participate in any $\sigma-$suboptimal path. These vertices are found in linear time by using dynamic programming to compute the highest scoring paths from the source vertex (vertex corresponding to mass 0) to each vertex v and from each vertex v to the sink (vertex corresponding to the precursor mass). After summing up these values and removing vertices for which the resulting score is deficient by more than σ, we are left with very small spectrum graphs (typically 50% of vertices are removed, leaving about 20 vertices per 1000 Da of mass). These filtered graphs contain a smaller number of paths, that can be generated by a depth-first search that prunes paths that cannot lead to $\sigma-$suboptimal solutions.

4.B.3 Peptide identification Using De Novo Sequences

Most database search algorithms follow a canonical approach in which the query spectrum is compared to every database peptide within a given mass tolerance. Precision mass spectrometry offers the opportunity to forgo this potentially time-consuming approach. We show how de novo sequencing enables a fast database search program that does not involve comparison of spectra to database peptides and has running time that is practically independent of the database size and the number of PTMs being searched.

In a sense, the approach we present below extends the idea of filtration [75, 219], by capitalizing on the high precision of FT-ICR to create longer and more accurate *gapped* tags. Our algorithm consists of two stages. In the first stage, we generate de novo peptide reconstructions which are used in the second stage, the database lookup.

Our algorithm works as follows. Given a query spectrum, we generate the top k de novo reconstructions (gapped peptides), as described in Section 4.B.2. Typically a value of $k = 10$ will suffice to have a 98% retrieval rate from the database (see Table 4.3).

We then proceed to use the gapped de novo peptide reconstructions for the database lookup. While searching the database with a gapped peptide P is already much faster than the spectrum vs. database scan performed by algorithms such as Sequest [61] or Mascot [162], we further speed up the search and forgo the database scan altogether. This is achieved by filling gaps in P with all possible combinations of amino acids and further searching database with the resulting set P^* of continuous amino acid strings. This can be done instantly if the database is preprocessed, such as using a hash table or suffix tree (checking if P^* is present in a hash table typically requires a single read to memory). Note that these indexed database need to be created only once, and this too can be done relatively quickly (creating a hash table for a large sequence file takes only several seconds).

In practice it does not make sense to query the database with peptides longer than 8 amino acids since spurious database hits of such length have negligent probability. Therefore if P^* contains sequences longer than 8 amino acids, we include in P^* sequences of length 8 that are generated from the sub-path of P with the minimal number of possibilities to fill its gaps. When the generated sequences do not span the entire mass range of the original peptide, we take note of the distance from the $N-$terminal to mass of the vertex at the beginning of the sequences' path, and the distance from the end of the path to the $C-$terminal similarly to InsPecT algorithm [219]. These mass offsets are very useful for filtering spurious database hits since most random hits to the database will not have flanking sequences that can lead to a successful extension to the correct N- and C-terminal masses.

4.C Results

4.C.1 MS/MS Data

Our data set contains 376 MS/MS spectra of doubly charged tryptic peptides that were generated by an Agilent 1100 nanoflow system coupled to a 7-tesla hybrid linear ion trap Fourier transform mass spectrometer (LTQ-FT, Thermo Electron Corp., Bremen, Germany), see ref [186] for further details on the experimental protocol. The spectra were pre-processed to remove isotopic peaks, and have relatively few noise peaks

(the average peak density was 30 peaks per 1000 Da of mass). All spectra were identified by Mascot [162] with high confidence, and had sufficient fragmentation to support a gapped peptide of at least 6 amino acids. The spectra belonged to peptides with lengths in the range 6-25 amino acids, with an average length of 11.1. Since the mass resolution of FT-ICR is very high, we used a mass tolerance of 0.0075 Da (i.e., we identify a peak if it falls within margin of 0.0075 Da from its expected position). Even with such a narrow tolerance 95% of the $b-$ and $y-$ ions that are present in the spectrum are identified. Such a narrow tolerance represents almost one hundred-fold improvement in resolution compared to regular ion-trap LTQ.

4.C.2 Fourier Transform Mass Spectrometry and Peptide Fragmentation

An investigation of our dataset reveals that FT-ICR can be used to gain new insights into peptide fragmentation. Since collision-activated dissociation (CAD) was performed by an LTQ mass spectrometer we expect to find the typical abundant fragments such as $y-$ and $b-$peaks and their derivatives [49, 73, 93, 213]. However, with FT-ICR it is possible to detect rare ion-fragments, which could not be identified with lower resolution instruments since they would be indistinguishable from noise (see for example analysis on similar data with low resolution instruments [101]). Therefore, instead of analyzing the data in the *validation* mode, where one tests whether the already known ion fragments are present in MS/MS spectra, we first analyzed our dataset in the *discovery* mode that allows one to discover new unsuspected fragment ions and evaluate their propensities. We used the *offset frequency function* [49], which finds recurring mass offsets in the spectra which help to identify the types of ion fragments that are present.

Table 4.2 lists fragment ions present in FT-ICR mass spectra and highlights the advantages of precision mass spectrometry: some of fragment ions in Table 4.2 are not detectable on standard instruments due to low signal-to-noise ratio. With such instruments the probability of observing a random noise peak is approximately 0.1 so most peaks would be virtually indistinguishable from the noise. All offsets included in the table have a probability which is much greater than the probability 0.001 of observing

Table 4.2: Information on ion types learned form 376 FT-ICR spectra of doubly charged peptides using the offset frequency function [49]. Note that the probability of observing a peak at random is 0.001. [(a)] the offset is relative to the mass of the respective prefix or suffix peptide (for doubly charged fragments, the offset is relative to half the mass of the prefix or suffix peptides). [(b)] the mass difference between the offset determined by the offset frequency function and the true mass of the fragment. [(c)] the number of observed fragment peaks vs. the number of possible positions at which the fragments could be detected. [(d)] the number of spectra which have at least 1 occurrence of the peak (maximal number 376). [(e)] These are "phantom" fragments due to harmonics of intense peaks [141].

Ion	Offset [(a)]	Δ [(b)]	# Peaks [(c)]	# Spectra [(d)]	Probability
y	19.020	0.002	2245/2792	376	0.804
b	1.006	-0.002	1934/2806	374	0.689
$b - H_2O$	-17.005	-0.002	777/2744	264	0.283
$y/2$ [(e)]	9.508	-0.001	508/2359	293	0.215
$y - H_2O$	1.005	-0.003	312/2360	211	0.132
y^{+2}	10.012	-0.001	316/2448	215	0.129
$b - NH_3$	-16.021	-0.002	253/2746	119	0.092
a	-26.988	-0.001	205/2706	144	0.076
$[y - H_2O]^{+2}$	1.006	-0.002	156/2246	127	0.070
$[y - H_2O - H_2O]^{+2}$	-7.998	0.000	142/2189	134	0.065
$b - H_2O - H_2O$	-35.015	-0.002	119/2661	60	0.045
$y - NH_3$	1.989	-0.003	110/2689	79	0.041
$[y - H_2O - NH_3]^{+2}$	-7.507	-0.001	75/2192	73	0.034
$b/2$ [(e)]	0.503	-0.001	64/2139	42	0.030
$b - H_2O - NH_3$	-34.031	-0.002	71/2663	42	0.027
$a - NH_3$	-44.015	-0.002	42/2652	38	0.016
$a - H_2O$	-44.999	-0.001	32/2650	25	0.012
$[y - NH_3]^{+2}$	1.498	-0.001	23/2248	20	0.010
b^{+2}	1.006	-0.002	14/2146	12	0.007
$b - NH_3 - NH_3$	-33.047	-0.002	17/2664	11	0.006
$y - H_2O - H_2O$	-17.007	-0.004	12/2673	11	0.005
$y - H_2O - NH_3$	-16.022	-0.003	10/2676	10	0.004
Internal$+H$	1.005	-0.003	227/10841	144	0.021
Internal$+H - H_2O$	-17.005	-0.002	125/10345	84	0.012
Internal$+NH_2 + H_2O$	34.027	0.002	112/11633	92	0.010

a noisy peak[3], so these offsets are likely to represent fragmentation products. We emphasize that all these ion-fragments can contribute to the ability of de novo algorithms to recover the correct sequence. Even phantom fragments[4] can help by identifying the

[3]The probability of observing a noisy peak is approximated by $\frac{\#\text{unexplained peaks} \times 2 \times \text{tolerance}}{\text{precursor mass}}$.

[4]FT-ICR detects some "phantom" fragments that appear due to harmonics. These fragments that

charge states of their singly charged counterparts. Additional information on the relative intensity rank of the fragment ions is relayed in Table 4.1.

Due to the data's high accuracy and resolution, we were able to identify many internal fragments in addition to the standard single fragmentation ion products. We can also use FT-ICR to automatically derive the "fragmentation rules" for internal ion fragments (e.g., $N-$terminal of Proline and Glycine turned out to be preferred cleavage sites involved in the formation of internal fragments.) Such fragments, which cannot be reliably identified by low resolution instruments, can play a role in the scoring and validation of peptide identifications.

4.C.3 Homeometric Peptides

We ran several experiments to evaluate the phenomenon of homeometric peptides. Figure 4.3 shows the results of an experiment in which 10000 random peptides of various lengths were generated and tested to see if they have homeometric peptides. Two mass tolerance settings were tested: 0.5 Da. which is typical for low resolution ion-trap instruments, and a narrower tolerance of 0.0075 Da. used with high resolution FT-ICR. Figure 4.3 shows that the larger the tolerance, the more likely the occurrence of homeometric peptides. Thus, while homeometric peptides are quite common with a large mass tolerance of 0.5 Da., Figure 4.3 shows an average 20-fold reduction in the number of homeometric peptides when the tolerance is narrowed to 0.0075 Da.

Homeometric peptides do not only complicate de novo sequencing, they also limit the ability of database searches to make confident identifications. We conducted simulations to test how homeometric peptides affect database searches (Homo Sapiens protein sequences from NCBI release 35 with 16.8M amino acids) under low and high precision settings (mass tolerances 0.5 and 0.0075 Da., respectively). We examined randomly selected peptides of various lengths and determined their distance from the other peptides in the database. Each peptide was compared with all other peptides in the database whose precursor mass was within a specified margin from the precursor mass of the original peptide. For the tolerance of 0.5 we used a precursor mass margin of 1 Da, which yielded on average 300000 database peptides, and with the tolerance

appear as double (or higher) charged fragments are an artifact of lower charged intense peaks [141].

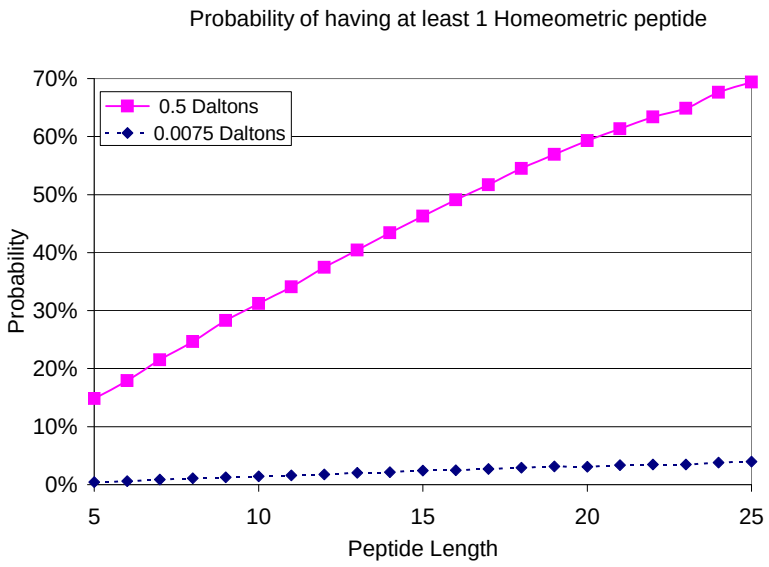


Figure 4.3: Probability of homeometric peptides. Random peptides of lengths 5-25 were generated and tested for the possibility of having at least one homeometric peptide (this test was done by generating their spectrum graphs and searching for multiple paths in the spectrum graph). Two mass tolerance settings were tested: 0.5 Da for low resolution and 0.0075 for high resolution.

of 0.0075 we used a precursor mass tolerance of 0.015 Da yielding an average of 4500 database peptides. Since in practice the mass spectra of a peptide P does not contain peaks from all the peptide's expected cleavages, we also report results for the peptide distances when the peaks of randomly selected cleavages were removed from P's set of expected masses (we report results for 0-4 missing cleavages).

Figure 4.4 presents the results for peptides of lengths 7, 14, and 21 amino acids. The top portion of the figure shows the results for low precision (tolerance 0.5 Da), and the bottom portion shows the results for high precision (tolerance 0.075 Da). Short peptides often have δ-homeometric peptides in the database for small δ (especially when the larger tolerance is used). The probability of having a homeometric peptide grows dramatically when some of the cleavages are missing. This explains scenarios in which database search tools cannot make conclusive identifications because there are several likely candidates (e.g., when Sequest [61] has several peptides with a high X_{corr}, but the resulting ΔC_n is low). Every pair of homeometric peptides creates a pair of

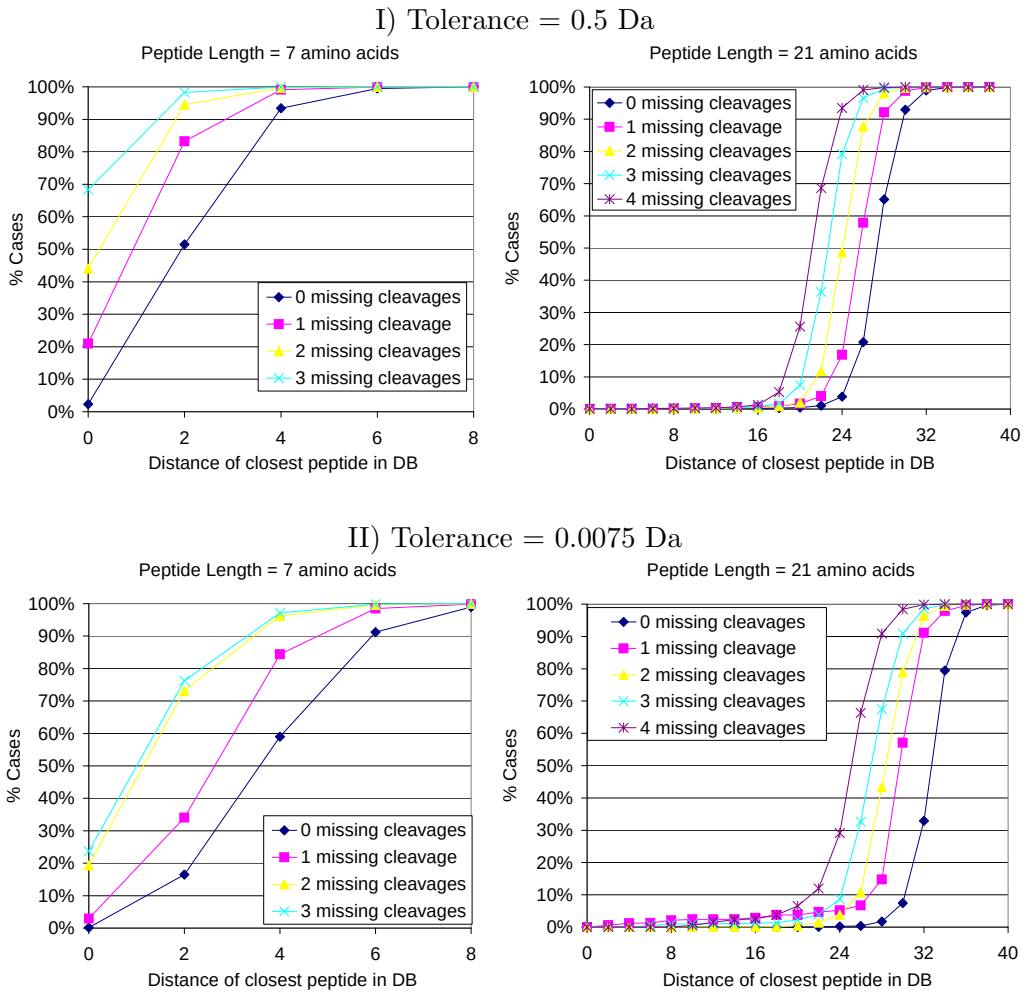


Figure 4.4: Probability that a database contains homeometric peptides. Random peptides were selected from a sequence database of 16.8 million amino acids and searched against the entire database to detect their closest δ-homeometric counterparts. Two tolerance settings where used: the top shows results for a tolerance of 0.5 Da, which models low precision data, the bottom shows results for a tolerance of 0.0075 Da, which models high precision data. The results are shown for random peptides of lengths 7 and 21, and for various numbers of missing cleavages.

"black holes" in the database - peptides that cannot be reliably identified even from high quality spectra. The probabilities of homeometric peptides in the high precision setting are significantly smaller. There are several reasons why this happens. First, the narrower tolerance restricts the creation of random spurious edges in the spectrum graph. In addition, using a narrow tolerance helps to resolve ambiguities due to the possible overlap of the integer masses of b- and y-ions, including overlap of their isotopic

Table 4.3: Correctness of De novo paths and number of generated peptides. The highest scoring de novo paths were generated for 376 mass spectra of doubly charged tryptic peptides, de novo path were generated. The table contains the percentage of spectra for which at least one of the k highest scoring paths ($k = 1, 5, 10$) is correct along with the number of unique peptides that were generated from those paths for the database lookup. The statistics are given for paths derived from spectrum graphs without PTMs, and for graphs containing 10 types of PTMs with the paths allowed to include only 1 or 2 PTM instances.

# De Novo	No PTMs		10 PTMs / 1 allowed		10 PTMs / 2 allowed	
Paths Used	% Correct	# Peptides	% Correct	# Peptides	% Correct	# Peptides
1	90.4	4.4	86.2	7.9	85.9	10.7
5	97.3	33.7	96.0	58.0	96.0	76.9
10	98.4	74.4	97.1	119.9	97.1	162.2

distributions. For instance, if monoisotopic masses are different by 1 or 2 Da, the overlap will still occur in low-resolution instruments, and the two ions will not be resolved. Finally, the narrower precursor mass tolerance means there are much fewer peptides in the database that have the potential to be homeometric (the number of these drops from 300000 with the precursor mass tolerance of 1 Da to 4500).

4.C.4 De Novo Sequencing with Precision MS

We ran de novo benchmark tests on our dataset of 376 spectra in order to evaluate our de novo algorithm's performance. Table 4.3 shows the probability that the set of k highest-scoring suboptimal paths contains the correct path. By considering more than a single path, the probability that a correct path was extracted grows from 90.4% using a single path, to 98.4% using 10 paths. The table contains statistics both for regular spectrum graphs (20 amino acids) and spectrum graphs that were constructed using 10 simulated PTMs (which effectively raises the number of amino acids used to construct the graph to 30). As customary [44, 47, 219] we restrict the number of PTMs in a peptide to either 1 or 2.

With the spectrum graphs that were constructed with 10 PTMs there are slightly lower success rates due to the larger number of edges that lead to more spurious paths. Naturally, there is a cost for considering more than a single de novo path and the tradeoff is an increase in the number of candidate peptides that need to be

Table 4.4: Peptides which were not covered by the 10 highest scoring paths. The table displays the true peptides, and the peptides corresponding to the highest scoring paths in the spectrum graphs, along with the number of their supporting peaks in the spectrum. The "." symbol represents a cleavage which has supporting peaks in the spectrum.

	Mascot Peptide		Top Ranked Path	
#	Peptide	# Peaks	Peptide	# Peaks
1	SI.A.V.S.L.PR	17/43	V.A.T.V.S.L.PR	20/43
2	GSL.GGG.FSS.G.G.F.S.G.GS.FSR	29/38	[314.17] .G.G.F.S.S.G.W.S.G. [1136.49]	27/38
3	RID.IT.L.S.S.V.K	10/37	[198.16] A.A.L.DMV.S.V.K	9/37
4	LAPITSD.P.TE.AT.A.V.G.A.V.EASFK	20/46	[394.27] .T.S.D.Q.HHP.A.V.Q.QT.LYR	13/46
5	IR.E.E.Y.PD.R	9/26	R.L.E.E.S.NSS.R	10/26
6	FNIS.N.G.G.PA.PE.AITDK	19/48	[373.21] .S.D.G.G.QKW.H.T. [1369.66]	21/48

looked up in the database (the larger the number of peptides that are used, the bigger the chance of having a spurious database hit.) While a single path, on average, generates 4.4 continuous peptide sequences, 10 paths generate 74.4 peptides. The number of peptides generated for paths from spectrum graphs with PTMs is higher since the PTMs offer more possibilities to fill the gapped paths. We remark that verifying 100 peptides against a database hardly leads to any increase in the overall running time as compared to matching a single peptide since the database is pre-indexed (e.g., with a hash table) and such matching takes a very small fraction of the overall running time.

It is worth mentioning that when compared to the results in Table 4.3, using our de novo approach on data from lower resolution ion-trap mass spectrometers (with a tolerance of 0.5 Da.), the results were much inferior. In a benchmark on a test set of ion-trap spectra of tryptic peptides [73] that did not consider PTMs, the top de novo path was only correct for 30% of the spectra, while the probability that one of the top 10 scoring paths was correct was only 52%. Such low accuracy would cause many missed identifications and therefore precludes the application of our novel peptide identification approach on data from low resolution instruments.

Table 4.4 shows all peptides for which the set of the top 10 highest scoring paths in the spectrum graph did not contain a correct path. These peptides point to a somewhat less reliable Mascot scores or even potential errors in original Mascot identifications. For instance, for peptide SIAVSIPR (first row), the top de novo reconstruction VATVSLPR, which comes from the protease trypsin, "explains" the spectrum significantly better than the Mascot database hit (de novo reconstruction explains 20 out of

Table 4.5: Expected number of random database hits and successful extensions. The table shows the expected number of times in which a single peptide sequence has a random database hit and a random hit that is successfully extended to obtain a complete peptide match. The experiments were run using a sequence database of 50 million amino acids. Data was collected for peptides of lengths 6-8 amino acids, and under two types of de novo searches, the regular search, and a search that considered 10 PTMs. The data was collected on a training set of 376 mass spectra of doubly charged tryptic peptides.

DB	Sequence	No PTMs		10 PTMs 1 allowed		10 PTMs 2 allowed	
Size	Length	E[#hits]	E[#ext]	E[#hits]	E[#ext]	E[#hits]	E[#ext]
50 M	6	2.1	0.0075	2.37	0.017	2.44	0.024
	7	0.139	0.0025	0.164	0.0054	0.174	0.0076
	8	0.012	0.0003	0.016	0.0007	0.018	0.0011

43 spectrum peaks, whereas the Mascot identification explains only 17 out of 43 peaks).

4.C.5 Random database hits and extensions

We first wanted to determine the feasibility of using de novo sequences for direct lookup in a database, in particular we wanted to determine how likely we are to have random database hits and successful extensions of the sequence to the N- and C-terminals. We tested our approach on the set of 376 test spectra described above. While these test spectra did not contain PTMs, we simulated searches that consider PTMs by adding the PTM edges to the spectrum graphs. Table 4.5 contains statistics on the tendency to have random hits and successful extensions with a large 50M database. When the spectrum contains peaks from a peptide that fragmented well and the generated de novo paths are quite long (≥ 8 amino acids), the chances of a random database hit become very low. The situation is different when the candidate sequences are short; they can generate several database hits for consideration. When such a hit is found, we attempt to extend it to a full match by finding in the database flanking sequences which match the prefix and suffix masses. Given the narrow mass margins that are tolerated with our data, it is unlikely that an incorrect database hit can be extended correctly.

Table 4.5 shows an approximate reduction of two orders of magnitude between the probability of a database hit and the probability of a successful extension of that hit (the reduction is higher for shorter peptides because they have a higher rate of extensions occurring simultaneously towards the $N-$terminal and the $C-$terminal.) When PTMs

are involved in the search, they offer more opportunities both for database hits and especially, many more possibilities to form correct extensions which is why the searches with PTMs have higher rates of false matches.

Table 4.5 also highlights some of the complications that occur dealing with short peptides (length 6-7 amino acids). Even with precision mass spectrometry, many algorithms cannot confidently identify them when searching a large database without using additional information (such as knowing that the protein in question had previous identifications with other mass spectra). In such cases it is advisable to minimize the probability of the algorithm returning a false identification. This can be done by either reducing the database size, using a small number of de novo reconstructions (possibly one), or limiting the search to non modified peptides.

4.C.6 Database Search

For the sake of simplicity we used a slightly naïve approach towards the implementation and testing of the database search. For each mass spectrum we used our de novo algorithm to generate a set P^* of amino acid sequences (as described in Section 4.B.3). The sequences P^* were sorted in a decreasing order of their de novo scores and submitted for database lookup in that order. The first sequence that had a database hit and could be successfully extended to the $N-$ and $C-$ terminals, was returned by the algorithm as the spectrum's identification (and the search terminated). If no such peptide was found, the algorithm terminated indicating that it could not find a peptide for the spectrum in the database.

Table 4.6 contains results of our benchmark experiments in which we applied the aforementioned procedure to our set of 376 spectra[5]. As could be expected , the more de novo reconstructions are used, the larger the proportion of correct identifications (true positives) since the set of de novo reconstructions is more likely to contain a correct sequence (see Table 4.3). Note that in any case, even a small set of 5 de novo reconstructions is sufficient for identifying correctly over 97% of the spectra. Since the database is searched with relatively long peptide sequences, there are very few spurious

[5]The benchmark experiments were conducted on a desktop PC with a 2.8 GHz Pentium D processor and 2 GB of RAM.

Table 4.6: Peptide identification results for 376 mass spectra. The experiments measured the success rate of our algorithm under different conditions: various sequence database sizes (0.5 million, 5 million, and 50 million amino acids), different numbers of de novo paths (1,5,10), and three types of searches (without PTMs, a search that simultaneously considers 10 types of PTMs but allows at most one modified amino acid in the peptide, and a search that considers 10 PTMs but allows up to two modified amino acids). The results are shown in terms of: TP - true positives (correct identifications made by the algorithm), FP - false positives (erroneous peptide identifications made the algorithm), and FN - false negatives (instances in which the algorithm did not return any peptide identification).

Decoy DB Size	# De Novo Paths	Search Type								
		No PTMs			10 PTMs / 1 allowed			10 PTMs / 2 allowed		
		% TP	% FP	% FN	% TP	% FP	% FN	% TP	% FP	% FN
0.5 M	1	0.904	0	0.096	0.862	0	0.138	0.859	0	0.141
	5	0.973	0	0.027	0.960	0.003	0.037	0.960	0.003	0.037
	10	0.984	0	0.016	0.971	0.003	0.026	0.971	0.003	0.026
5 M	1	0.904	0	0.096	0.857	0.005	0.138	0.854	0.005	0.141
	5	0.971	0.003	0.026	0.952	0.013	0.035	0.949	0.016	0.035
	10	0.981	0.003	0.016	0.960	0.013	0.026	0.955	0.019	0.026
50 M	1	0.888	0.019	0.093	0.862	0.045	0.093	0.851	0.045	0.104
	5	0.952	0.021	0.027	0.920	0.059	0.021	0.915	0.056	0.029
	10	0.963	0.021	0.016	0.920	0.059	0.021	0.920	0.059	0.021

hits. However, the larger the database being searched, the larger the proportion of false positives we observe. This increase is due to spurious database hits of de novo reconstructions with a higher score than the correct sequence's score. It is likely that a less naïve approach that implements validation of the results via a scoring function would eliminate many of these false positives.

4.D Conclusion

Precision mass spectrometry, such as FT-ICR, opens the door to improved proteomics analysis and novel algorithms. For instance, with the increased mass resolution of FT-ICR we were able to detect many more types of fragment ions that would typically be statistically indistinguishable from noise with lower resolution ion-trap instruments. Even more important is the fact that precision MS helps to eliminate problems that hinder the analysis of data from low resolution instruments. We explored the phenomenon of homeometric peptides (different peptides with nearly identical sets of $b-$ and

$y-$peaks) that severely limits de novo sequencing with low precision data. With high precision data homeometric peptides are extremely rare, making peptide sequencing accurate. There have been recent computational techniques that can solve the problem of homeometric peptides by separating b- and y- ladders using a combination methods such as correlating between MS^2 and MS^3 spectra (Zhang and McElvain, 2000 [245]), or using complementary fragmentation techniques, such as CAD and ECD (Savitski et al., 2005 [186].) Bern and Goldberg, 2005 [17] used an optimization approach aimed at achieving this separation, while Bandeira et al., 2006 [12] used pairs of spectra (e.g., from a modified and unmodified version of the same peptide) to separate b- and y-ladders. Our analysis above shows that in most cases the high accuracy and resolution of FT-ICR alone can eliminate most of the problems caused by homeometric peptides, without the need for additional data required by previous approaches [12, 186, 245].

In this work we demonstrated the feasibility of a new approach to database search which relies on direct lookup of sequences in the database, as opposed to the standard methodology that compares a query mass spectra to peptides from a database. Even using a naïve approach to validation of search results, our method was bale to identify correctly 96% of the test spectra when searching a 50MB database. Our algorithm uses rapid de novo sequencing and replaces the traditional database scan with a direct sequence lookup in a pre-indexed database. It is capable of rapidly identifying peptides even when searching large databases and considering PTMs. The high precision of FT-ICR is necessary for our method's success, since de novo peptide sequencing with low precision data is not accurate enough.

Our approach can be viewed as an extremely efficient database filtration method. Previous filtration approaches to MS/MS database search used only short sequence tags (typically 3 amino acids long), so they need to consider many database hits and select the best one [33, 75, 139, 147, 212]. However, our predicted de novo sequences are much longer, so they have very few spurious hits in the database. Thus most of the database comparison in our method amounts to the evaluation of a single database hash hit since typically only the de novo sequence representing the correct peptide will have a database match. Our benchmark results demonstrate the feasibility of using de novo sequencing of precision MS data as the key component for a database search. The high accuracy

of the de novo sequencing leads to a very small fraction of missed identifications. Since there is a very low rate of spurious database hits, there will not be many false database hits competing with the correct hit, which can simplify the task of a scoring function to determine the single correct hit.

The idea of peptide identification by means of sequence lookup can be expanded to scenarios that are not addressed adequately with the current database search tools, such as identifying peptides that are products of alternative splicing or fused genes. A simple method for identifying such peptides could be to split each de novo sequence $S = s_1 s_2 \dots s_n$ into pairs of the form $S' = s_1 \dots s_k$ and $S'' = s_{k+1} \dots s_n$, and to lookup S' and S'' in the database. Finding hits for S' and S'' in different proteins can raise the possibility that the query spectrum belongs to a peptide that is a product of fused genes, while finding hits for S' and S'' in the same protein can indicate that the peptide is a product of alternative splicing.

Our de novo sequencing algorithm typically requires 0.05 seconds per spectrum. Since the peptide identification relies heavily on the de novo stage, its runtime scales well when the database size is increased and PTMs are added to the search. For instance, while searching against a 0.5M database without considering PTMs takes about 0.06 seconds per spectrum, this grows to approximately 0.2 seconds per spectrum when searching against a 50M database and considering 10 different PTMS. This 3-fold increase in runtime is much smaller than the more than 100-fold increase that would be incurred by traditional database search programs, whose runtime typically increases linearly with the increase in database size and exponentially with the number of PTMs simultaneously considered. Having run-time that is practically independent of database size is essential for an efficient implementation of the advanced database searches such as the ones described above. The effective database size being searched can grow dramatically if one wants to consider all possible peptides that could be products of alternative splicing or fusion proteins (the latter effectively squares the number of peptides that need to be considered). The traditional approaches which compare spectra to database sequences would incur a hefty increase to the run-time due the extreme growth in the effective database size, while our novel approach which relies on hash table sequence lookups would be much more resilient.

Error tolerant homology searches [89, 192, 194, 195] are another avenue through which we can benefit from the high performance of de novo sequencing of precision mass spectrometry data. Due to the high rate of de novo sequencing errors encountered with low precision data, there are many cases in which matches are missed by such algorithms because the de novo sequences vary too much from the spectrum's correct peptide, even though that peptide (or a close homologue) are present in the searched database.

The accurate de novo sequencing of precision mass spectrometry data can also be used to flag spectra for further investigation. For instance, if a spectrum returns no database hit but has high scoring de novo reconstructions, it is very likely that the spectrum belongs to a real peptide that is not present in the database. In this case, we can use the set of de novo sequences, which with a very high probability contain a variant that is completely correct, to look for alternative explanations for the source of the spectrum (e.g., instances of alternative splicing or fusion proteins, as described above).

This chapter, in full, was published as "De Novo Peptide Sequencing and Identification with Precision Mass Spectrometry". A.M. Frank, M.M Savitski, M.L. Nielsen, R.A. Zubarev, and P.A. Pevzner. *Journal of Proteome Research*, 6:114-123, 2007. The dissertation author was the primary author of this paper.

5

Predicting Fragment-Ion Peak Ranks

5.A Introduction

Analyzing the large volume of MS/MS data that is being generated these days raises formidable computational challenges. However, this surplus of data also opens a window of opportunity, enabling us to use advanced data-driven machine learning methods to improve the quality of our analysis. Machine learning methods have been used extensively in recent years for solving problems in involved in MS/MS analysis. Such problems include scoring peptide-spectrum matches [8, 25, 35, 38, 39, 59, 64, 73, 93, 118, 162, 219, 242], spectral quality assignment [19, 70, 150], precursor charge determination [37, 122], validation of search results [224], prediction of proteotypic peptides [138, 215], retention time prediction [7, 124, 166], and more.

In this chapter and the next one, we explore how the boosting algorithm [79] (in the context of ranking [40, 77]) can leverage the large amounts of experimental data to derive powerful models that offer superior solutions for challenging problems in computational mass spectrometry. Our first ranking-based application is to create a predictor for peak fragment ranks, which makes its decisions solely according to the peptide's amino acid sequence. Our second application, described in Chapter 6, is a ranking-based scoring function for peptide-spectrum matches, that uses the peak rank

predictions, along with many other types of features, to greatly enhance the accuracy of sequencing algorithms.

5.A.1 Classification vs. Ranking

Machine learning deals with algorithms that enable a computer to "learn" from data (inductive learning). A common machine learning task is the classification of data instances. Let $\mathcal{X}$ be a set called the domain or *instance space*, and let $\mathcal{Y}$ be a finite set of *class labels*. In a *supervised learning* setting, the machine learning algorithm is given a set of n labeled training data $(x_1, y_1), \ldots, (x_n, y_n)$, where $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. The learning algorithm's objective is to derive a model that is successful at assigning the correct class $y \in \mathcal{Y}$ when given new, previously unobserved, instances $x \in \mathcal{X}$. There have been many popular machine learning approaches developed for this task such as Neural Networks (the Perceptron algorithm [179]), Naïve Bayesian classifiers [54], Support Vector Machines (SVMs) [190, 226], to name a few. Given their popularity, classification algorithms are used frequently to solve a verity of problems.

Though many problems can be constructed as classification problems, the classification framework does not always provide the optimal solution. This is especially true for problems whose inherent structure suggests using other frameworks. For example, queries to internet search engines may return many webpages as answers. Usually one cannot state that an answer to a query is completely right or completely wrong, rather a more common approach is to assign a degree of relevance to each returned webpage. In such cases, instead of using a classification algorithm to dichotomize the results, we would prefer a *ranking* algorithm which scores the answers on a gradient presenting the most relevant answers first (when the ranking algorithm is used to refine a previous ordering, it is also called *reranking*).

Ranking algorithms have been used for several machine learning tasks such as collaborative-filtering of search engine results [77, 108, 161], combining expert opinions in recommender systems [1, 77], and several natural language processing applications [40, 41, 42]. However, despite their effectiveness, ranking algorithms have not drawn much attention from practitioners. We demonstrate that ranking algorithms can be used effectively to solve challenging problems in computational mass spectrometry.

We hope this work will encourage others to try these methods to solve similarly structured problems.

5.A.2 The RankBoost Algorithm (Freund et al., 2003)

Following is a brief summary of the RankBoost algorithm of Freund et al. [77], which is the algorithm we use in the next two chapters for our ranking-based applications. This summary is produced here in order for the manuscript to be self-contained, and to allow us to discuss the specific details concerning our use of the algorithm. For a complete description of the algorithm, including practical usage examples and theoretical analysis (error bounds, etc.), see the aforementioned reference.

A Formal Framework for the Ranking Problem

The goal of the RankBoost learning algorithm is to produce an ordering of a set of elements given to it from an instance space $\mathcal{X}$. The learning algorithm achieves this by combining a given set of preferences, or rankings, of the instance space. We use the term *ranking feature* to denote these given rankings of the instances. A ranking feature f is an ordering of the instances from most preferred to least preferred. We can equivalently think of f as a scoring function where higher scores are assigned to more preferred instances. We do not require that all instances be ordered by every ranking feature. Formally, a ranking feature f is a function $f : \mathcal{X} \to \mathbb{R} \cup \{\perp\}$, where $f(x) = \perp$ denotes the fact that f does not assign a score to instance x (for example because the feature is not applicable to the instance x). In such cases we say f "abstains" on x.

The final ranking is obtained by combining the scores given to instances by the individual ranking features. The final ranking has the same form as that of the ranking features: it gives a linear ordering of the instances (with ties allowed). However, unlike ranking features, the final ranking does not abstain on any instances. Formally, the final ranking is a function $H : \mathcal{X} \to \mathbb{R}$, with a similar interpretation to that of the ranking features, i.e., x_1 is ranked higher than x_0 by H if $H(x_1) > H(x_0)$.

The RankBoost learning algorithm performs its training by observing the ordering of pairs of instances from the instance space. For a pair of instances $x_0, x_1 \in \mathcal{X}$, we denote the fact that x_1 is ranked higher than x_0 by using the tuple (x_0, x_1). During

the training phase, we supply the learner with training data that consists of ordered pairs of elements $(x_0, x_1) \in \mathcal{X} \times \mathcal{X}$. This information is given in the form of a *feedback function* $\phi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The fact that $\phi(x_0, x_1) > 0$ means that x_1 should be ranked above x_0. Similarly, $\phi(x_0, x_1) < 0$ denotes that x_0 should be ranked above x_1. A value of $\phi(x_0, x_1) = 0$ means that there is no preference regarding the relative ranking of x_0 and x_1. The magnitude $|\phi(x_0, x_1)|$ denotes how important it is to rank x_1 above or below x_0. We further assume that $\phi(x, x) = 0$ for all $x \in \mathcal{X}$, and that ϕ is anti-symmetric in the sense that $\phi(x_0, x_1) = -\phi(x_1, x_0)$ for all $x_0, x_1 \in \mathcal{X}$. If $\phi(x_0, x_1) > 0$ we say that the pair (x_0, x_1) is *crucial*.

The feedback function ϕ can be used to define a distribution over the training data. Let $D(x_0, x_1) = c \cdot \max\{0, \phi(x_0, x_1)\}$. Thus, all negative entries of ϕ (which carry no additional information) get set to zero. The value of the positive constant c is chosen so that $\sum_{x_0, x_1 \in \mathcal{X}} D(x_0, x_1) = 1$.

The learning algorithms that we study attempts to find a final ranking H with a small weighted number of crucial-pair misorderings. This quantity is called the ranking loss and is denoted $\text{rloss}_D(H)$ and is given by

$$rloss_D(H) = \sum_{x0, x1 \in \mathcal{X}} D(x_0, x_1) \, [\![H(x_1) \leq H(x_0)]\!] = Pr_{(x_0, x_1) \sim D} \left[H(x_1) \leq H(x_0)\right], \quad (5.1)$$

where we define $[\![\pi]\!]$ to be 1 if predicate π holds and 0 otherwise.

A Boosting Algorithm for the Ranking Task

The learning algorithm for the ranking problem described here is based on a machine learning method called boosting [80, 188]. Boosting is a method of producing highly accurate prediction rules by combining many "weak" rules which may be only moderately accurate. In the current setting, we use boosting to produce a function $H : X \to \mathbb{R}$ whose induced ordering of $\mathcal{X}$ approximates the relative orderings encoded by the feedback function ϕ. The boosting algorithm for the ranking is called RankBoost; its pseudocode is shown in Figure 5.1.

Like all boosting algorithms, RankBoost operates in rounds. We assume access to a separate procedure called the weak learner that, on each round, is called to produce a weak ranking. Weak rankings have the form $h_t : X \to \mathbb{R}$. RankBoost creates these

RankBoost

Given: initial distribution D over $\mathcal{X} \times \mathcal{X}$.
Initialize: $D_1 = D$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak ranking $h_t : \mathcal{X} \to \mathbb{R}$. (see Section 5.A.2)
- Choose $\alpha_t \in \mathbb{R}$.
- Update: $D_{t+1}(x_0, x_1) = \frac{D_t(x_0,x_1)exp(\alpha_t(h_t(x_0)-h_t(x_1)))}{Z_t}$
  where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final ranking: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

Figure 5.1: RankBoost algorithm of Freund et al. [77].

ranking hypotheses by converting the continuous ranking features into a step function (see below). RankBoost maintains a distribution D_t over $\mathcal{X} \times \mathcal{X}$ that is passed on round t to the weak learner. Intuitively, RankBoost chooses D_t to emphasize different parts of the training data. A high weight assigned to a pair of instances indicates a great importance that the weak learner order that pair correctly. The value of α_t is set using the third method described in ref [77] (this is a heuristic method that works for weak rankings with range $[0, 1]$, and is based on minimizing Z). Each round of boosting requires running time $O(|\phi| + n \cdot |\mathcal{X}_\phi|)$, where $\mathcal{X}_\phi = \{x \in \mathcal{X} | \exists x' \in \mathcal{X} : \phi(x, x') \neq 0\}$. The final ranking H is a weighted sum of the weak rankings, and has a bound on its loss: $rloss_D(H) \leq \prod_{t=1}^{T} Z_t$. We refer to the sum $H(x)$ as the *ranking score* given to x by H.

Finding Weak Learners

The RankBoost algorithm requires access to a *weak learner* to produce weak rankings (also called *weak hypotheses*). RankBoost's weak learners are {0,1}-valued weak rankings based on ordering information provided by the ranking features, but ignore specific scoring information (the benefits of using this format opposed to using the the

feature values themselves are explained in ref. [77]). The weak rankings take the form

$$h_i(x) = \begin{cases} 1 & \text{if } f_i(x) > \Theta \\ 0 & \text{if } f_i(x) \leq \Theta \\ q_{\text{def}} & \text{if } f_i(x) = \perp \end{cases} \tag{5.2}$$

Where $\Theta \in \mathbb{R}$ and $q_{\text{def}} \in \{0, 1\}$. That is, a weak ranking is derived from a ranking feature f_i by comparing the score of f_i on a given instance to a threshold Θ. The weak ranking assigns a default score q_{def} to instances left unranked by f_i. Selecting the optimal feature f_i and values of Θ and q_{def} can be done efficiently using dynamic programming [77].

5.B Methods

5.B.1 MS/MS Datasets

Our experiments with ranking algorithms were all conducted using a large set of approximately 320000 unique peptide-spectrums pairs collected from various MS/MS experiments involving low-resolution CID ion-trap mass spectrometers. Most of the data used was collected from the Briggs lab at UCSD (samples of from human HEK293 cell culture [218] and samples from *Dictyostelium discoideum* [217]), and the Smith lab at PNNL (samples taken from *Shewanella oneidensis* MR-1 [87, 143]).

We used the InsPecT database search tool [219] to perform peptide identifications (release 20070613), using the default search parameters (precursor mass tolerance 2.5 Da, fragment ion tolerance 0.5 Da). All searches were performed using a shuffled decoy database [10, 60, 97]. The InsPecT F-score threshold values for accepting identifications were selected to ensure a true positive peptide identification rate of 98% (i.e., only 2% of the peptide hits came from the decoy database).

A peptide's charge and parent mass greatly influence the nature of the experimental mass spectrum observed for a given peptide. For example, with a typical doubly-charged peptide of length 10 we can expect to see most of the peptide's b/y ions, including peaks belonging to cleavages that are one and two amino acids away from the peptide's terminals. However, with a large triply-charged peptide 30 amino acids long, we expect to see only a small portion of the b/y peaks, along with many doubly-charged

Table 5.1: MS/MS training dataset. The set of 319578 pairs of unique peptides and MS/MS spectra was partitioned according to charge and parent mass. For each partition we list its parent mass range, the number of peptides that fell in that range, and the typical length of those peptides (the lengths listed cover at least 95% of the peptides in each partition).

Charge 1			Charge 2			Charge 3		
Parent masses	#Unique peptides	Typical lengths	Parent masses	#Unique peptides	Typical lengths	Parent masses	#Unique peptides	Typical lengths
0-1150	20971	7-12	0-1100	25709	7-12	0-1950	13198	10-19
1150-1400	18984	9-15	1100-1300	33167	9-14	1950-2450	13131	16-24
1400+	16231	11-20	1300-1600	45595	10-16	2450-3000	12684	20-29
			1600-1900	43054	13-20	3000+	13824	25-48
			1900-2400	43225	15-25			
			2400+	19805	20-32			
56186			210555			52837		

b^{+2}/y^{+2} fragment ions. Due these differences it is better to train separate specific models for the different classes of peptides, rather than generate one general model for all peptides. To enable the generation of multiple models we divided the training data according to the partitioning described in Table 5.1. For each partition, we note the charge and parent mass range of its peptides, along with the number of peptides and their typical lengths. The number of partitions generated for each charge depended on the number of training peptides available (we divided the doubly-charged peptides into 6 parts, while the singly-charge were only divided into 3 parts). Partitioning the data this way also gave computational benefits; many of the partitioned models required up to several days to complete training and also used up most of the available RAM on the machines in the process.

A peptide's mass and charge are not the only factors that influence its mass spectrum. The peptide's amino acid composition also has a great influence on the fragmentation (see Section 1.C). In particular, the number and location of the basic amino acids (namely arginine and lysine) play an important role in influencing which peptide fragmentation mechanisms are most active. To reflect the important role amino acid composition plays in peptide fragmentation, we applied an additional partitioning of the training data according to proton mobility. Following the division suggested by Kapp et al. [115], we placed peptides in three categories, according to their amino acid composition:

- **Mobile peptides** - the number of basic residues (i.e., combined arginine, lysine,

and histidine residues) is less than the peptide's charge.

- **Nonmobile peptides** - number of arginine residues is greater than or equal to the peptide's charge.
- **Partially mobile peptides** - all peptides not classified as mobile or nonmobile.

This additional division according to proton mobility is only used the task of predicting peak ranks. In this task we only end up comparing peaks belonging to the same peptide, so they always get ranked according to the same model. With the other scoring tasks, such as ranking de novo sequences (described in Chapter 6), we can end up comparing peptides with different mobility states, so in that case, we only partition the models according to charge and parent mass, and avoid using different models for scoring peptides against the same spectrum.

5.B.2 Implementation of RankBoost Algorithm

We implemented the RankBoost algorithm in the C++ programming language. Our implementation largely follows the steps described in Section 5.A.2, however we added a few extra procedures designed to accelerate the model's convergence and reduce instances of *overfitting* (cases where the model's parameters are optimized to produce the best results on the training data, but are generally suboptimal for new unseen data).

Binning Some ranking features assign real values to instances. With large training sets this can impede the learning process. The procedure that selects weak learners needs to consider each of these possible $|\mathcal{X}|$ values (see Section 5.A.2), which can greatly increase the time required for each of the algorithm's iterations. A natural method to circumvent this issue is to employ binning, i.e., to partition the feature values observed in the training set into B bins and use a single value to represent the values in each bin. Applying this step reduces the time needed to find the optimal weak learner among n features from $O(n|\mathcal{X}|)$ to $O(nB)$. Furthermore, binning can help reduce instances of overfitting, since it reduces the ability of the algorithm to optimize weights to suite a small number of instances. We typically use $B = 50$ bins for real-valued features.

Dual threshold search for weak learners The implementation of the weak learner selector described in ref. [77] finds a single threshold Θ for each feature, assigning 0 to values below Θ, and 1 to values above it. However, our experiments have shown that the models converge much quicker if the weak learner is allowed to choose two thresholds Θ_1, Θ_2 to enable a weak learner of the form

$$h_i(x) = \begin{cases} 1 & \text{if } \Theta_1 < f_i(x) \leq \Theta_2 \\ 0 & \text{if } f_i(x) \leq \Theta_1 \text{ or } f_i(x) > \Theta_2 \\ q_{\text{def}} & \text{if } f_i(x) = \perp \end{cases} \tag{5.3}$$

Since we restrict the number of possible values a feature can take to a relatively small B, the quadratic nature of this search does not become computationally prohibitive.

Restricting the optimization to selected features The weak learner selection procedure as described in ref. [77] considers all features in each iteration. However, the nature of the domain we deal with is that it has a fixed number of features to consider, and does not necessarily use all of them. For example, the peak ranking models described in Section 5.C can use over 800 features, but even after 10^5 rounds, typically less than 500 are deemed informative and are incorporated in the model. Therefore, starting from round $t = 100$, our algorithm mostly only considers a subset of "active" features that are already included in the model. Every so often the algorithm will have an iteration in which it considers all features, so features from the excluded set have an opportunity to be added to the active set of features in the model. Using this procedure was found to reduce the running time required for the models' training to converge.

Condensing weak learners During the T rounds of the model training, features can get selected to be used in weak learners multiple times (i.e., the same feature f_i is used to create $h_{t_1}, h_{t_2}, \ldots, h_{t_k}$). We condense these multiple weak learners into a single *cumulative weak learner* h_f^*

$$h_f^*(x) = h_{t_1}(x) + \ldots h_{t_k}(x). \tag{5.4}$$

This accelerates the algorithm's execution, since calculating the score $H(x)$ is done in in time linear in the number of features n, and not in the much larger number of rounds

T. Using cumulative weak learners is also convenient for plotting the scores given by the rank model for different feature values $f(x)$. Figure 5.4 gives an example of a cumulative weak learner after various numbers of feature weight updates.

Regularization of instance weights To avoid assigning excessive importance to incorrect training examples we restricted the weights training samples can achieve in a manner similar to LogitBoost [82].

5.C RankBoost Models For Predicting Peak Ranks

The reproducible nature of mass spectra makes it possible to use simple but powerful comparison-based methods like spectral libraries for peptide identification [46, 81, 125, 131, 204, 239]. While this method is accurate, it misses identifications of uncommon peptides (PTMs, products of miscleavages, peptides from rarely expressed proteins or alternative splice variants, etc.) To be able to identify all spectra using a similarity based method, one would have to be able to predict a *theoretical* mass spectrum for any peptide sequence. The predicted theoretical spectrum must be sufficiently similar to the observed spectra in order for the similarity based identification to succeed. The popular Sequest algorithm [61], which relies on the cross-correlation between the theoretical spectrum of a database peptide and the experimental mass spectrum, skirts the issues involved in predicting theoretical spectra by relying on a naïve fragmentation model that assigns a fixed equal intensity to the major ions (b/y) and a lower fixed intensity to their neutral losses. In practice, most experimental spectra are not very similar to their Sequest-predicted theoretical spectra, which limits Sequest's identification capabilities [114].

Developing more accurate statistical models for peptide fragmentation is essential for improving peptide identification algorithms [14, 101, 142, 213, 234]. Several programs have begun to use detailed statistical models of peptide fragmentation in their scoring [14, 35, 59, 73, 191, 219]. Recently, Zhang [243, 244] described a kinetic model that simulates the peptide fragmentation process in order to predict theoretical MS/MS spectra from peptide sequences. This model is also used as the basis for a score for a

Peak rank prediction problem

Input:

- Peptide sequence $P = p_1 p_2 \dots p_n$, where p_i, $1 \leq i \leq n$, are amino acids.
- Set of fragment ion types $\mathcal{F}$, e.g., $\mathcal{F} = \{b, y, a, y - H_2O, b^{+2}, \dots\}$.

Output:

- A permutation π of the set of all possible fragment peaks ($\mathcal{F} \times \{1, \dots, n\}$), where π is ordered according to decreasing intensity (e.g., $\pi = y_8, y_6, b_3, b_4, b_3 - H_2O, \dots$).

Figure 5.2: The peak rank prediction problem.

database search program that outperformed Sequest and Mascot [207]. However, Zhang's kinetic models are quite complex and involve multi-iteration simulations to produce a theoretical spectrum. The prediction process also becomes less accurate and more computationally intensive with large peptides and higher charge states [244]. This can limit its applicability for advanced high-throughput proteomics such as searching large databases (many candidates to consider) or peptides with PTMs (which are not currently simulated in the kinetic model).

Due to the complexity of peak intensity prediction, at this stage we abstain from developing a model for an accurate theoretical spectrum (one that would be sufficient for a identification by a simple spectrum similarity test). Instead, we undertake a less ambitious goal of developing a model for predicting the peaks' relative ranks in a spectrum (using ranks instead of peak intensities is something we also argue for in the case of de novo scoring, see Section 4.B.2 and ref. [74]).

The problem we solve is called the *peak rank prediction problem* and is outlined in Figure 5.2. In this problem, we are given a peptide sequence $P = p_1 \dots p_n$ and a set of fragment ions $\mathcal{F}$, and we need to predict π, which is a ranked permutation of P's fragment ions. The goal is for the permutation π to resemble, as much as possible, the peak ordering observed in an experimental mass spectrum of the peptide P. Since the weaker peaks' intensities are usually below the instrument's detection level, not all n peaks for a fragment type are likely to be observed. Therefore, we are mainly concerned with the predicted ordering of the strongest peaks.

5.C.1 Feature functions for peak rank prediction

Our peak rank models rely solely on information that can be extracted from a peptide sequence $P = p_1 \dots p_n$. For each peptide cleavage position $1 \leq i < n$ and fragment type $f \in \mathcal{F} = \{y, b, \dots\}$, we fill a separate feature vector using the feature functions described below. These vectors are scored by the model, and then we use the rank scores to induce a permutation. For example, if b_4's rank score is higher than y_7's, this is interpreted as the fact that we expect b_4's intensity to be higher than y_7's in the experimental spectrum. We do not deduce from this additional information, such as by how much b_4's intensity is higher, or what is the ratio between the two fragments intensities.

There are several different types of sequence-based features that are predictive of a peaks intensity. Below we describe the different classes of features and give examples of of some feature function values in the trained models. The total number of features that can be assigned to each fragment type is over 200. In our trained models, we included in $\mathcal{F}$ only the four most abundant fragment types (typically $b, y \in \mathcal{F}$). This means that our models can potentially have over 800 features. However, in practice, not all features get included in the models since they do not bring a significant improvement to the ranking performance compared to the features already selected to be in the model (features are only added to a model if they reduce the ranking error [77]).

Even though a model might contain hundreds of features, the actual feature vectors that are created for peaks are extremely sparse (typically less than 10-20 feature functions do not abstain, see Figure 5.6 for an example). This happens because most of the features abstain on any given peak instance (for example, only the features belonging to y-ions participate in the scoring of a y-peak, the rest of the features abstain). In addition, many of the features, especially the amino acids adjacency features below, come in sets of 20, for which only one feature is assigned a nonzero value.

The first feature we use is an indicator function which holds the type of fragment being examined (e.g., $b, y, b - H_2O$, etc.) This feature is used to give a prior score for the different fragment types (usually the b- and y-ions receive a positive score while the neutral losses get zero).

Following is a description of the three additional types of features we use: peak location features, adjacent amino acid features, and peptide composition features.

Peak location features

The relative mass of a peak has a significant influence on the peak's intensity. Peaks located near the center of the peptide generally have higher intensity than the peaks near the N- and C-terminals [213]. Peaks near the terminals are often close to the edges of the"visible" range of masses which the instruments are calibrated to detect[1], and can thus suffer from instrumental bias that records a weaker signal for them. Another possible reason for this location bias, is that the cleavage of the amide bonds might be more energetically favorable near the center of the peptide, than it is near the terminals.

Let Min be the minimal peak mass that can be detected with a peptide of P's precursor mass, Max be the maximal mass that can be detected (the minimum between the precursor mass and 2000), m be theoretical mass of the peak being ranked, and i ($1 \leq i < n$) be the index of the cleavage location being examined. There are 3 features (per fragment ion type $t \in \mathcal{F}$) used to describe the location of a peak:

- $f^t_{rel}(m) = \frac{m-Min}{Max-Min}$ - The relative location of the peak with mass m in the range [Min,Max].
- $f^t_{min}(m) = m - Min$ - The distance from the minimal detected mass.
- $f^t_{max}(m) = Max - m$ - The distance from the maximal detected mass.

Note that for each instance, only one of the features $f^t_{min}(m)$ and $f^t_{max}(m)$ gets assigned a value depending on wether the peak is closer to the N-terminal or the C-terminal. The superscript t in the feature names reflects the fact that there is a separate feature created for each fragment ion type $t \in \mathcal{F}$.

Adjacent amino acid features

The identity of the amino acids adjacent to a cleavage site play an important role in the determining the propensity of the peptide to undergo fragmentation at that

[1]Usually mass spectrometers are calibrated to detect peaks between a minimal mass that is approximately $\sim 0.25\cdot$ precursor mass, and a fixed maximal mass of 2000 Da. For example, if the peptide has a precursor mass of 2500 Da, only b and y peaks that have a mass of 625-2000 Da can be detected.

site [23, 101, 115, 213]. These influences can be very significant, such is case with the proline effect [23, 225], where the peaks that are products of a cleavage N-terminal to proline are extremely strong.

We derived a simple set of pattern-based features to describe the amino acids adjacent to a cleavage site. We only look at the amino acids that are up to 3 positions away from the cleavage site, since amino acids farther away do not have as strong an influence on the fragmentation. In total, we created 6 sets of 20 features each, which we describe below. Let X denote one of the 20 standard amino acids, $P = p_1 \dots p_n$ be the peptide sequence for which we want to predict peak ranks, i be the index of the cleavage site we are examining (i.e., i is the number of amino acids that are on the N-terminal side of the cleavage position), and $t \in \mathcal{F}$ be the fragment type being scored. The indicator function $I[\![p_i = X]\!]$ returns 1 if the amino acid p_i is X and 0 otherwise. Our 6 sets of features can be expressed using the following indictor functions:

- $f^t_{Cut-3=X}(P, i) = I[\![p_{i-2} = X]\!]$ - The amino acid 3 positions before the cleavage site is X.
- $f^t_{Cut-2=X}(P, i) = I[\![p_{i-1} = X]\!]$ - The amino acid 2 positions before the cleavage site is X.
- $f^t_{Cut-1=X}(P, i) = I[\![p_i = X]\!]$ - The amino acid directly before the cleavage site is X.
- $f^t_{Cut+1=X}(P, i) = I[\![p_{i+1} = X]\!]$ - The amino acid directly after the cleavage site is X.
- $f^t_{Cut+2=X}(P, i) = I[\![p_{i+2} = X]\!]$ - The amino acid 2 positions after the cleavage site is X.
- $f^t_{Cut+3=X}(P, i) = I[\![p_{i+3} = X]\!]$ - The amino acid 3 positions after the cleavage site is X.

These are sparse features; for any given peptide and cleavage index i, at most 6 of the 120 features takes a value 1. Less than 6 features might take a nonzero value if the index i being examined is near the terminals (e.g., $i = 1$). In that case, it makes no sense to examine position $i - 2$.

Peptide composition features

Besides examining the amino acids adjacent to cleavage site, we can also extract important information from a peptide's general amino acid composition. For instance, if a peptide has basic amino acids near the N-terminal, rather than the C-terminal, the observed b-ion fragments can be stronger than the y-ion fragments, contrary to what is typically observed with tryptic peptides. Furthermore, there are amino acids that are known to be more prone to neutral losses, such as asparagine which often loses an NH_3. A simple way to capture this sequence information is to count the number of occurrences of each amino acid on both sides of the cleavage site. We created the following two sets of 20 features for each fragment type $t \in \mathcal{F}$:

- $f^t_{N\#X}(P,i) = \sum_{j=1}^{i-3} I[\![p_j = X]\!]$ - The number of times X appears in P on the N-terminal side of the cleavage.
- $f^t_{C\#X}(P,i) = \sum_{j=i+4}^{n} I[\![p_j = X]\!]$ - The number of times X appears in P on the C-terminal side of the cleavage.

Note that we exclude the amino acids that are 1-3 amino acids away from the cleavage site, since these are covered by the adjacent amino acid features.

We also take note of the amino acids on the N- and C-terminals, since these specific positions can have a special influence on the fragmentation outcome. We have two additional sets of features to express this information:

- $f^t_{N=X}(P) = I[\![p_n = X]\!]$ - The amino acid on the N-terminal is X.
- $f^t_{C=X}(P) = I[\![p_n = X]\!]$ - The amino acid on the C-terminal is X.

Creating Training and Validation Sets

In order to train a model for with RankBoost, we need to supply the algorithm with two types of data: samples from the instance space $\mathcal{X}$ and a feedback function $\phi : \mathcal{X} \times \mathcal{X}$. For each partition of the training data according to Table 5.1, we first separate the peptides into three sets based on their proton mobility (mobile, partially mobile, and nonmobile, see page 85). For each training set of peptides, we repeat the following process:

- **Selecting fragment types to be modeled** - We examine the set of peptides, and assign the four most abundant fragment ions to the model's fragment set $\mathcal{F}$ (usually $b, y \in \mathcal{F}$). The model being created only predicts ranks for these fragment peaks.

- **Adding instances to $\mathcal{X}$** - For a peptide $P = p_1 \dots P_n$, we compute the masses of the $4 \cdot (n-1)$ non-trivial peaks corresponding ion types $\mathcal{F}$, ignoring the ones that fall outside of the range detected by the mass spectrometer. We record each of these $k \leq 4n-4$ fragments' actual intensity in the spectrum (or assign it zero if it has no peak in the spectrum). For each peak i, $1 \leq i \leq k$, we create a feature vector x_i^P using to the ranking features described above, and add x_i^P to the instance space $\mathcal{X}$.

- **Adding pairs to feedback function ϕ** - Let $x_1^P, \dots, x_k^P$ be the k instances created for P's fragment peaks, and let I_i be the intensity observed in the mass spectrum for fragment peak i. Since the feedback function only requires pairs (x_i^P, x_j^P) in which $I_j > I_i$, we exclude all pairs where peak $I_j \leq I_i$. Let $I_{\max} = \max\{I_{i_1}, \dots, I_{i_k}\}$. For each pair (x_i, x_j), we assign the following value to the feedback function ϕ:

$$\phi(x_i^P, x_j^P) = \frac{I_j - I_i}{I_{\max}}. \tag{5.5}$$

When we examine repeated MS/MS experiments involving the same peptide, we often observe an element of randomness in the peak intensities. Pairs of low-intensity peaks often have different ranks in different experimental spectra. Therefore, with low intensity peaks, the order observed is often determined not so much due to mechanisms that govern peptide fragmentation, but more due to the randomness of the intensity measurements. Applying the weighting scheme in Eq. 5.5 limits the influence of these questionable ordered instance pairs on the models being trained.

During the training we assess the model's performance, by measuring its rank loss (Eq. 5.1, page 82). Measuring the performance on the same set of data used to train the model is not recommended since this usually leads to model overfitting. To avoid this pitfall, we set aside a portion of the data to serve as a *validation set* which is used to test the performance of the model (and determine what configuration of the

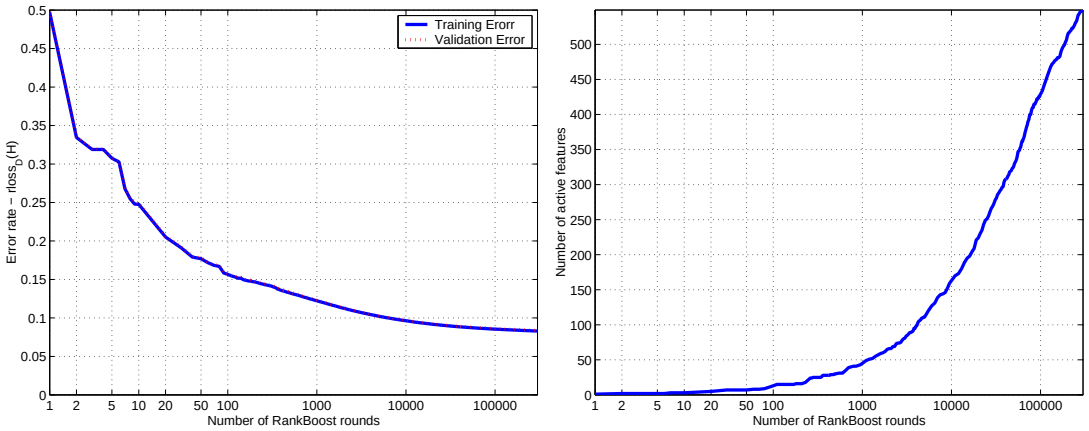


Figure 5.3: Statistics of RankBoost training for 300000 rounds. the left-side graph displays the training and validation error rates after running the RankBoost algorithm for a given number of rounds, and the right-side graph displays the number of features included in the model (i.e., the features that have a nonzero wight). The x-axis displays the number of boosting rounds using a logarithmic scale. The figures were generated using a training set of doubly-charged "mobile' peptides with precursor masses 1100-1300.

parameters is optimal in general). We typically set aside 1/4 of the training data for validation purposes.

5.C.2 Training the RankBoost Models

A typical training set we used to train the models consisted of 10000 peptides, which created ≈ 300000 instances (peaks) in $\mathcal{X}$, and $\approx 3 \cdot 10^6$ pairs of the form $(x_0, x_1) \in \mathcal{X} \times \mathcal{X}$ in ϕ. Our algorithm is capable of performing approximately 2 rounds of boosting per second on this data using a single CPU. The training typically required between 100000 to 400000 rounds to complete. We now examine in detail the training of a specific model for predicting fragments in doubly-charged "mobile" peptides of mass 1100-1300 Da, in order to observe the dynamics of the convergence of the RankBoost model.

Figure 5.3 depicts the statistics obtained from the training of the ranking model. The greatest reduction in error rates is achieved in the initial boosting rounds. The first feature to be selected in the models is f^y_{rel} (Relative peak location of y). Using this feature reduced the initial error from about 50% to 33%. By round 10 there are only 4 features in the model, and they alone reduce the ranking error to 26.6%. After 100

rounds, the model includes 17 features, the training error was reduced to 15.1%. From this stage on, adding more features and performing additional rounds of boosting yields a small but steady improvement in the model's performance. After 100000 rounds, using 423 features, the training error was 7.97%. At the stage of the model's convergence at round 200000, the validation error is minimal (7.89%), and the model includes 512 features. Continuing with additional rounds does not improve the validation error, but does start to widen the gap between the validation and training error (to a maximum of 0.08%). This gap, though small, is evidence that the model is experiencing overfitting, and being optimized specifically for the training data.

Figure 5.4 illustrates the dynamics of tuning the scores of weak learners associated with a ranking feature. The figure shows a graphical representation of the cumulative weak learner (described in Section 5.B.2) of the feature f^y_{rel} (relative peak location of y), after various numbers of update rounds. Each plot shows the cumulative scores for different ranges of $f^y_{rel}(P, i)$ between 0, designating that the peak is located at the minimal detectable mass, and 1, designating a peak at the maximal detectable mass. After the first round the hypothesis that was most beneficial was to assign a score +0.5 if the y peak falls between 5%-80% of the mass range. Subsequent update rounds refine the function further, decreasing the score near the extremities and increasing it near the center. This feature's behavior corresponds to a known phenomenon observed with MS/MS spectra of peptides, in which the peaks near the center are, on average, much stronger than the peaks detected in the low and high mass ranges. These figures also illustrate the typical behavior of the learning algorithm, which makes large changes in the initial rounds, but only smaller changes later on.

In Figure 5.5 we compare the feature f^y_{rel} with peak location features for the other fragment types used in the model: b, $b - NH_3$ and $b - H_2O$. All four features show a similar trend, the scores near the terminals are much lower than the scores near the center of the detected mass range. However, the magnitude of the b and y feature scores is much greater than the magnitude observed for $b - NH_3$ and $b - H_2O$. This reflects the fact that b- and y-ions are typically much stronger (and higher ranked) than ions with neutral losses.

Table 5.2 lists the most positive and most negative adjacent amino acid features

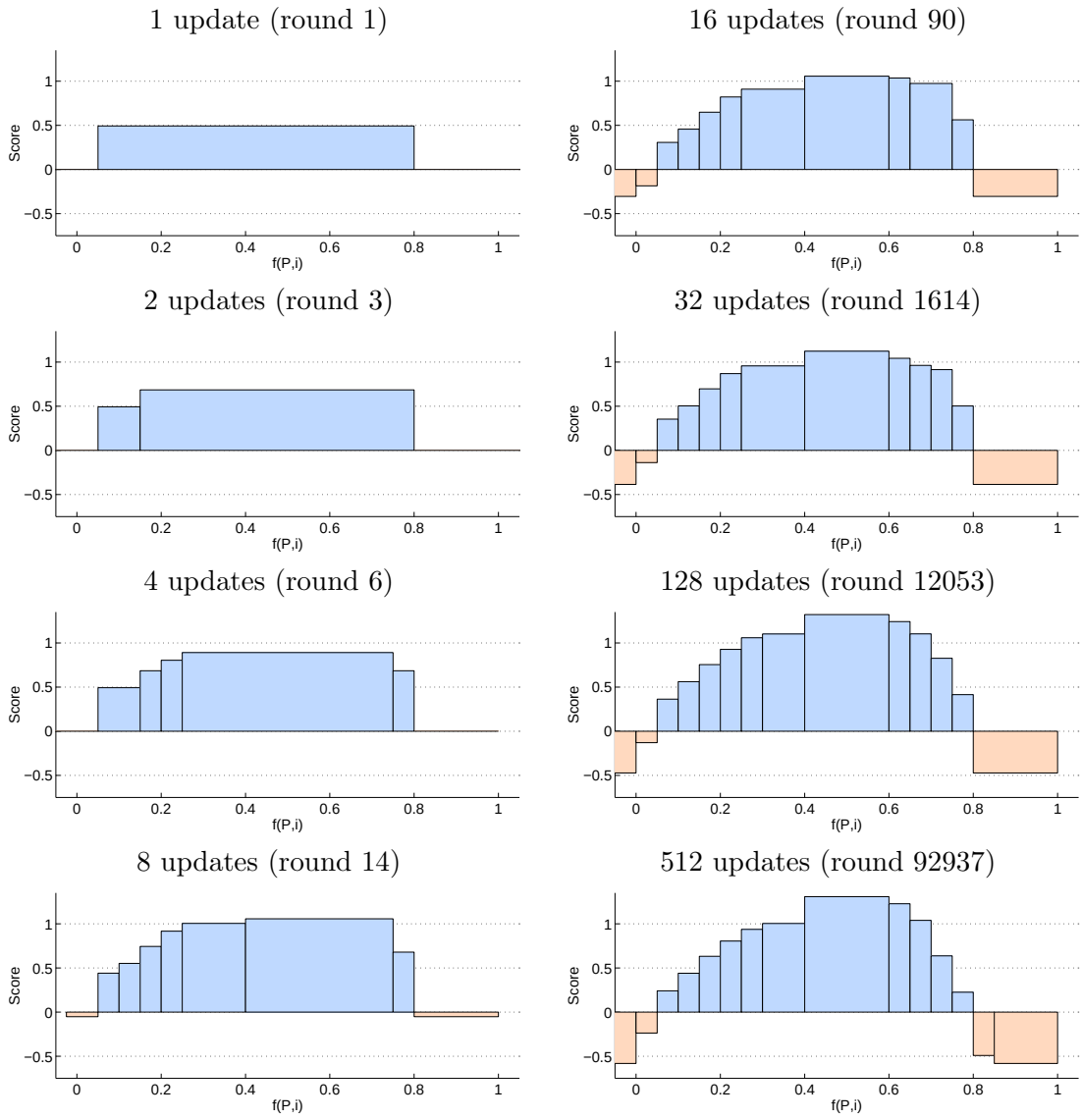


Figure 5.4: Updating Relative y-ion peak location feature. The figure shows the weights assigned to the feature describing the y-ion peak's relative location (the x-axis values that are between 0 and 1) at various update stages (after 1,2,4,8,16,32,128, and 512 updates). The values on the x-axis are binned into 20 bins of equal length. The y-axis holds the scores given by the model to the different peak relative locations.

in the model for the $y, b, b - NH_3$, and $b - H_2O$ ions, trained on doubly charged mobile peptides of mass 1100-1300 Da. In the interest of clearer presentation the names of the features were shortened, omitting $f^t_{(...)}(P, i) = 1$ (since all features in the same column belong to the same fragment type). Note that with the features listed in the table, as with all our indicator function-based features, we assume that if the indicator returns a

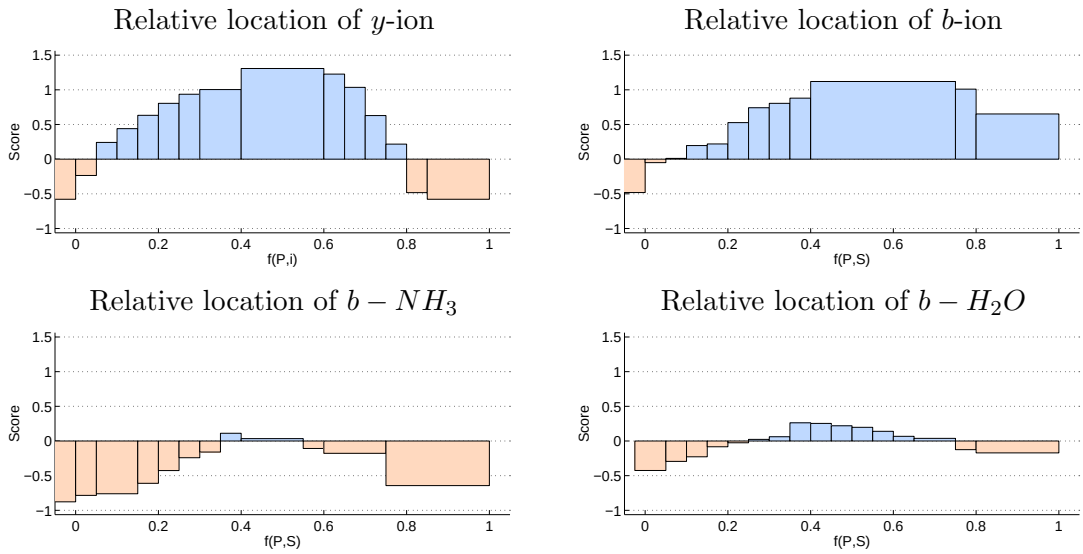


Figure 5.5: Comparison of peak location features for different ion types. The figure shows the scores given to the same feature function (relative peak location) for different fragment ion types: $y, b, b-NH_3$, and $b-H_2O$. The x-axis represents the peak's relative location. The y-axis holds the scores given by the model to the different peak relative locations. The different magnitudes of the scores reflect the fact that on average, b and y-ions are ranked much higher than $b - NH_3$ and $b - H_2O$.

value 0, the feature's score is also 0.

The features listed capture much of the previously observed phenomena regarding the influence of amino acids on peptide fragmentation [101, 115, 212]. With mobile peptides, the most active fragmentation pathways tend to be charge-directed, and most prominent among these is the cleavage N-terminal to proline. This is reflected in the tables with the highest scoring feature for all fragments being "Cut +1 = P". The influence of proline goes beyond the position directly N-terminal to the cleavage site, since "Cut +2 = P" and "Cut +3 = P" also have high scores. When the proline is on the other site of the cleavage the resulting peaks tend to be extremely weak, and often they do not get detected. Consequently, the features "Cut -1 = P" tend to have negative scores. Glycine (G) is known to have a similar effect to proline (though often weaker). Interestingly, glycine does not have strong positive score when it appears N-terminal to the cleavage, but does have a large negative score when it is C-terminal to the cleavage, which is greater than the score for proline with the $y, b - NH_3$ and $b - H_2O$ fragments.

The tables also show other known effects, such as the positive influence of the

Table 5.2: Most positive and negative adjacent amino acid features. The features are taken from the model for the $y, b, b-NH_3$, and $b-H_2O$ ions, trained on doubly charged mobile peptides of mass 1100-1300 Da.

Most positive features

y		b		$b-NH3$		$b-H2O$	
Feature	Score	Feature	Score	Feature	Score	Feature	Score
Cut +1 is P	1.52	Cut +1 is P	1.37	Cut +1 is P	0.95	Cut +1 is P	1.09
Cut −1 is V	0.79	Cut +2 is P	0.78	Cut +2 is P	0.56	Cut +2 is P	0.68
Cut −1 is L	0.68	Cut −1 is V	0.69	Cut −3 is N	0.53	Cut −1 is V	0.65
Cut +2 is P	0.57	Cut −1 is L	0.57	Cut −1 is V	0.50	Cut −1 is L	0.52
Cut −2 is L	0.50	Cut −2 is L	0.43	Cut −2 is N	0.39	Cut −2 is T	0.48
Cut +3 is H	0.45	Cut −2 is W	0.33	Cut −1 is L	0.35	Cut −2 is S	0.36
Cut +1 is G	0.44	Cut −2 is V	0.26	Cut +3 is P	0.17	Cut −1 is W	0.33
Cut −2 is W	0.37	Cut −1 is W	0.25	Cut −2 is W	0.13	Cut −1 is A	0.29
Cut −2 is V	0.36	Cut −2 is A	0.24	Cut +3 is H	0.13	Cut −1 is T	0.25
Cut −2 is A	0.32	Cut +3 is P	0.24	Cut −1 is Q	0.12	Cut +3 is P	0.20

Most negative features

y		b		$b-NH3$		$b-H2O$	
Feature	Score	Feature	Score	Feature	Score	Feature	Score
Cut −1 is G	-1.24	Cut −1 is P	-1.07	Cut +1 is R	-0.45	Cut −1 is N	-0.45
Cut −1 is P	-1.08	Cut +1 is R	-0.98	Cut −1 is G	-0.38	Cut +1 is R	-0.40
Cut −1 is S	-0.67	Cut −1 is G	-0.82	Cut −1 is S	-0.28	Cut −1 is G	-0.34
Cut −1 is N	-0.64	Cut +1 is H	-0.72	Cut −2 is E	-0.27	Cut −2 is C	-0.28
Cut −3 is P	-0.48	Cut −3 is G	-0.67	Cut −1 is P	-0.23	Cut −1 is P	-0.27
Cut −3 is Y	-0.46	Cut +2 is H	-0.64	Cut −2 is D	-0.21	Cut −2 is E	-0.21
Cut −3 is G	-0.46	Cut −3 is S	-0.63	Cut −1 is D	-0.19	Cut +1 is Q	-0.21
Cut −3 is T	-0.44	Cut −3 is T	-0.59	Cut +1 is D	-0.15	Cut −2 is D	-0.19
Cut −3 is S	-0.41	Cut −1 is S	-0.58	Cut +1 is E	-0.14	Cut +1 is E	-0.17
Cut −1 is D	-0.40	Cut +2 is K	-0.58	Cut −3 is Y	-0.11	Cut +1 is D	-0.15

aliphatic amino acids (e.g., A,V,L) when they appear N-terminal to the cleavage. There are also known negative effects such as when the amino acids serine (S) or threonine (T) appear N-terminal to the cleavage, or when histidine (H) is C-terminal to the cleavage [102, 115]. The table also shows amino acid influences that are particular to specific fragment ions. For instance serine and threonine that are known to promote water loss [159, 184], have positive scores in $b-H_2O$, but not with the other fragment types. Similarly, glutamine (Q) and asparagine (N) are known to increase loss of ammonia which explains their features' positive scores with $b-NH_3$. Another interesting phenomenon is the negative influence that the acidic amino acids (D,E) have on the intensity of neutral losses, especially $b-NH_3$, which seems to be more prominent than the effect they

Table 5.3: Adjacent amino acid features for b-ion in peptides of different mobility states. The features are taken from the models trained on doubly charged mobile peptides of mass 1100-1300 Da.

Most positive features

Mobile		Partially mobile		Nonmobile	
Feature	Score	Feature	Score	Feature	Score
Cut +1 is P	1.37	Cut −1 is D	1.01	Cut −1 is D	1.30
Cut +2 is P	0.78	Cut +1 is P	1.00	Cut +1 is P	0.77
Cut −1 is V	0.69	Cut −1 is H	0.75	Cut −2 is H	0.63
Cut −1 is L	0.57	Cut −2 is H	0.70	Cut −3 is R	0.63
Cut −2 is L	0.43	Cut −3 is R	0.62	Cut −1 is E	0.62
Cut −2 is W	0.33	Cut −1 is V	0.42	Cut −1 is H	0.62
Cut −2 is V	0.26	Cut +2 is P	0.37	Cut −1 is R	0.40
Cut −1 is W	0.25	Cut −1 is E	0.37	Cut −1 is V	0.28
Cut −2 is A	0.24	Cut −1 is L	0.31	Cut +2 is P	0.23
Cut +3 is P	0.24	Cut +1 is L	0.19	Cut −1 is L	0.23

Most negative features

Mobile		Partially mobile		Nonmobile	
Feature	Score	Feature	Score	Feature	Score
Cut −1 is P	-1.07	Cut −1 is P	-0.91	Cut −1 is P	-0.73
Cut +1 is R	-0.98	Cut −1 is G	-0.72	Cut −1 is G	-0.71
Cut −1 is G	-0.82	Cut −1 is S	-0.62	Cut −1 is S	-0.49
Cut +1 is H	-0.72	Cut −1 is T	-0.47	Cut −1 is T	-0.46
Cut −3 is G	-0.67	Cut +1 is R	-0.35	Cut +1 is R	-0.38
Cut +2 is H	-0.64	Cut −1 is N	-0.32	Cut +2 is R	-0.27
Cut −3 is S	-0.63	Cut −3 is G	-0.30	Cut +1 is D	-0.25
Cut −3 is T	-0.59	Cut +1 is D	-0.29	Cut −3 is P	-0.24
Cut −1 is S	-0.58	Cut −3 is P	-0.29	Cut −1 is N	-0.23
Cut +2 is K	-0.58	Cut −2 is P	-0.26	Cut −3 is G	-0.17

have on the b- or y-ions. The tables also show the negative effect of having arginine on the C-terminal side of the cleavage ("Cut +1 = R") with the $b, b - NH_3$ and $b - H_2O$ fragments. This can be explained by the fact with doubly charged mobile peptides, if there is an arginine on the C-terminal side, that means that there is no additional one on the N-terminal side. This sequesters most of the charge on the suffix fragments, which makes the prefix b fragments much weaker. In addition, the fact that this feature is most negative at the +1 position means that the arginine itself also interferes with the fragmentation (this fact was also previously observed [102, 115]).

Table 5.3 compares the amino acid adjacency features for b-ions across different mobility states (mobile, partially mobile, and nonmobile). The main difference between these models is the role that the charge-remote pathways play (in particular the fragmentation that is initiated by having aspartic acid (D) on the N-terminal side of the cleavage site). With the mobile peptides, the most positive feature is "Cut +1 = P", while "Cut -1 = D" is not even among the top ten. This starts to change with the partially mobile peptides, where both features have similar strong scores. However, with the nonmobile peptides, the charge-remote pathways become more dominant, which is manifested in the table by a much higher score for "Cut -1 = D". Another difference, between mobility states is the positive effect of having histidine N-terminal the cleavage site (e.g., "Cut -1 = H"). Note that these feature do not appear in the mobile model because doubly charged mobile peptides with a tryptic end cannot contain additional basic amino acids like histidine (see definitions on page 85). When examining the most negative features, we see that the lists for the different mobility states are quite similar.

Table 5.4 examines the most positive and most negative peptide composition features in the model for doubly charged mobile tryptic peptides with mass 1100-1300 Da. The strongest positive features involve the counts of basic amino acids on both sides of the cleavage site. For the y-ion, the highest scores are given to the presence of basic amino acids on the C-terminal side of the cleavage, while for the b-ion the presence of basic amino acids on the N-terminal side is rewarded. Interestingly, having histidine on the C-terminal is also highly rewarded with the y-ion (the score for having at least one histidine is 0.78). This might be because having histidine in the vicinity generally increases the fragmentation at the site [102]. The most positive features for the b-ions involve the presence of basic amino acids on the N-terminal side of the cleavage site. The presence of aliphatic amino acids (such as leucine and alanine) is also rewarded with high scores. For $b - NH_3$ we note that as expected, having asparagine (N) on the N-terminal side of the cleavage is highly rewarded (asparagine has a tendency to lose NH_3). In addition, the positive score for having multiple phenylalanines can be explained by the known fragmentation mechanisms that involve loss of NH_3 from aromatic amino acids [58, 128]. As far a the negative features are concerned, all feature types are penalized for having proline on the N-terminal side of the cleavage. This reflects the far reaching effects of

Table 5.4: Most positive and negative peptide composition features. The features are taken from the model for the $y, b, b-NH_3$, and $b-H_2O$ ions, trained on doubly charged mobile peptides of mass 1100-1300 Da (only features for the b, y and $b-NH_3$ ions are shown). In all the features we assume that amino acid counts of 0 receive score 0.

Most positive features

y		b		$b-NH3$	
Feature	Score	Feature	Score	Feature	Score
#C-side H > 0	0.81	#N-side H > 0	1.23	#N-side N $=$ 1	0.46
				#N-side N $=$ *more*	0.74
#N-side H > 0	0.78	#N-side L $=$ 1	0.30	#C-side P $=$ 1	0.10
		#N-side L $=$ 2	0.48	#C-side P $=$ *more*	0.45
		#N-side L $=$ *more*	0.73		
#C-side K > 0	0.67	#N-side K > 0	0.61	#N-side Q $=$ 1	0.09
				#N-side Q $=$ *more*	0.26
#N-side K > 0	0.39	#N-side A $=$ 1	0.19	#C-side R > 0	0.25
		#N-side A $=$ 2	0.34		
		#N-side A $=$ 3	0.50		
		#N-side A $=$ *more*	0.60		
#C-side R > 0	0.36	#N-side R > 0	0.55	#N-side F $=$ 1	0.02
				#N-side F $=$ *more*	0.21

Most negative features

y		b		$b-NH3$	
Feature	Score	Feature	Score	Feature	Score
#N-side P $=$ 1	-0.33	#C-side Q $=$ 1	-0.11	#N-side P $=$ 2	-0.16
#N-side P $=$ 2	-0.59	#C-side Q $=$ *more*	-0.28	#N-side P $=$ *more*	0.01
#N-side P $=$ *more*	-0.54				
#C-side Q $=$ 1	-0.20	#N-side P $=$ 1	-0.14	#N-side S $=$ 1	-0.04
#C-side Q $=$ *more*	-0.48	#N-side P $=$ 2	-0.10	#N-side S $=$ *more*	-0.05
		#N-side P $=$ *more*	0.01		
#C-side A $=$ 1	-0.06	#N-side T $=$ 1	-0.10	#N-side D $=$ 1	0.01
#C-side A $=$ 2	-0.20	#N-side T $=$ 2	-0.09	#N-side D $=$ *more*	-0.05
#C-side A $=$ *more*	-0.35	#N-side T $=$ *more*	-0.03		
#C-side L $=$ 1	-0.10	#C-side L $=$ 1	-0.04	#N-side V $=$ 1	-0.01
#C-side L $=$ 2	-0.17	#C-side L $=$ 2	-0.07	#N-side V $=$ *more*	-0.04
#C-side L $=$ *more*	-0.24	#C-side L $=$ *more*	-0.10		
#C-side N $=$ 1	-0.12	#N-side G $=$ 1	-0.03	#C-side L $=$ 2	-0.03
#C-side N $=$ *more*	-0.22	#N-side G $=$ 2	-0.06	#C-side L $=$ *more*	-0.02
		#N-side G $=$ *more*	-0.09		

proline (as seen in Table 5.3).

Table 5.5 lists the most positive and most negative N- and C-terminal amino acid features. The strongest features for the y-ion add premiums when the C-terminal is basic ("C-term aa is R" and "C-term aa is K"). There is also a high premium when the N-terminal is proline. This is most likely given to counter the negative scores assigned for the amino acid adjacency features like "Cut -1 = P", since when a proline is near the terminals its positive and negative effects on fragmentation are diminished. Interestingly, the highest positive terminal feature scores belong to "N-term aa is Q" and "N-term aa is W" for $b-NH_3$ and "N-term is E" for $b-H_2O$. These feature scores are a manifestation

Table 5.5: Most positive and negative terminal amino acid features. The features are taken from the model for the $y, b, b-NH_3$, and $b-H_2O$ ions, trained on doubly charged mobile peptides of mass 1100-1300 Da.

Most positive features

y		b		$b-NH3$		$b-H2O$	
Feature	Score	Feature	Score	Feature	Score	Feature	Score
N-term is P	0.69	N-term is P	0.19	N-term is Q	1.01	N-term is E	1.10
C-term is R	0.50	N-term is D	0.13	N-term is W	0.70	N-term is Q	0.10
C-term is K	0.12	C-term is F	0.12	N-term is Y	0.26	C-term is G	0.04
N-term is M	0.12	N-term is G	0.11	N-term is E	0.21	N-term is D	0.03
N-term is T	0.10	N-term is T	0.07	C-term is K	0.01	N-term is V	0.01

Most negative features

y		b		$b-NH3$		$b-H2O$	
Feature	Score	Feature	Score	Feature	Score	Feature	Score
N-term is E	-0.20	N-term is E	-1.06	N-term is S	-0.09	N-term is P	-0.05
N-term is Q	-0.10	N-term is Q	-0.81	N-term is T	-0.06	C-term is K	-0.04
N-term is D	-0.08	C-term is R	-0.50	N-term is P	-0.04	N-term is W	-0.04
N-term is G	-0.02	C-term is K	-0.49	N-term is N	-0.03	N-term is T	-0.02
N-term is N	-0.02	C-term is H	-0.30	N-term is D	-0.01	N-term is L	-0.02

of the glutamine/glutamic acid effect [90], in which extensive loss of NH_3 occurs when the N-terminal amino acid is glutamine, and a loss of H_2O when the amino acid is glutamic acid. Our results also indicate that there is substantial loss of NH_3 when tryptophan is on the $N-$terminal (though involving a different fragmentation mechanism [129]). The most outstanding negative scoring features are "N-term is Q" and "N-term is E" with the b- and y-ions. This is also due to the glutamic acid/glutamine effects. When the N-terminal amino acid is either glutamine or glutamic acid, the scores of b- and y-ions get reduced in order to increase the ranks of the $b-NH_3$ and $b-H_2O$, respectively.

5.C.3 Experimental Results

We trained a total of 39 models for peak rank predictions (3 mobility states $\times$ 13 partitions described in Table 5.1). The experiments below test the accuracy of these models for the peak rank prediction task.

First we examine how peak ranks get predicted in practice. Figure 5.6 gives a detailed example of the calculation of the peak rank scores of the top three peaks in the peptide GEEVTPISAIR. Part *a* displays the experimental mass spectrum, in which

the top three peaks are y_6 (intensity 89.9), y_7 (intensity 67.5), and b_5 (intensity 45.9). Part b of the figure holds a table that shows how the peak rank scores are calculated (only features that have a nonzero score are listed). The calculated rank scores induce the same rank ordering that is observed in the experimental spectrum (y_6 has a score of 4.67, y_5 has a score of 4.22, and b_5 has a score of 3.22). These scores were computed using the model for mobile doubly charged peptides of mass 1100-1300 (this model's most prominent features are listed in Tables 5.2-5.5). In this model, both the y- and b-ion fragment types receive a score of +0.43 ($b-NH_3$ and $b-H_2O$, which are the other two fragments in the model, receive a score of 0). The table also shows that the peak location features give a large score premium to all three peaks since they are b- and y-ions situated near the center of the peptides. Generally, the adjacent amino acid features have a lesser weight than the position features, with exception of the features "Cut -1 = P" which gives a high score premium of +1.52 for y_6 and +1.37 to b_5 (note that y_6 and b_5 get different scores since they are different fragment ions, and are assigned different features: $f^y_{Cut-1=P}$ and $f^b_{Cut-1=P}$, respectively). The peptide composition features generally contribute low scores, with exception of the "# C-side R=1" which gives +0.36 to the y-ions. The major factor that contributes to raising the y-ions scores' compared to b_5's is the feature "C-term aa is R" which gives +0.5 to the y-ions and −0.5 to the b-ion. This feature can be explained by the fact that with a mobile peptide, if the C-terminal is R, this means that most of the charge is sequestered at the C-terminal which leads to strong y-ions and weaker b-ions.

In order to examine how well the peak rank predictions fit experimental spectra, we selected a test case of peptides of length 10 and ran benchmark experiments with them. Figure 5.7 shows histograms of the ranks predicted for the peaks observed in the experimental spectra with ranks 1,3,5, and 10. The figures show that the rank predictions for the stronger peaks are much more accurate than the rank predictions for weaker peaks. For example, in over 60% of the cases the peak predicted to be strongest was in fact the strongest peak in the spectrum. However, when we examine the tenth strongest peak in the spectrum, we see that only in 9% of the cases, the predicted rank is correct. In addition, with the stronger peaks (ranks 1 and 3), the predicted ranks tend to be more concentrated around the observed rank, while with the weaker peaks (ranks

5 and 10), the predictions are more spread out.

We tested the peak prediction models on peptides of various lengths and charges. Since it is difficult to make exact rank predictions, we examined the proportion of cases in which the predicted rank p was at most 3 positions away from the observed rank r (i.e., $r - 3 \leq p \leq r + 3$). Table 5.6 holds results of these experiments with peptides of various lengths with charges 1,2, and 3. There are a few general trends we can observe in the table. First, as noted above, the predictions for stronger peaks are more accurate than weaker peaks (this is evident in all cases were the accuracy deteriorates as we move from rank 1 to rank 7). In addition, the peak rank predictions with shorter peptides are generally more accurate than the ones done with longer peptides. However, the decline in prediction accuracy is not that severe. For instance, there is only a 5.6% reduction in the accuracy of the prediction of the strongest peak when go from doubly-charged peptides of length 10 (92.9% correct) to peptides of length 20 (87.3% correct); even though the number of peak ranks that need to be predicted is doubled. The table also shows that it is generally more difficult to predict ranks in triply-charged peptides, than it is for doubly or singly-charged. This is most likely due to the more complex fragmentation processes that occur in triply-charged peptides which generally result in poor fragmentation that is difficult to predict.

5.D Discussion

In this chapter we explored how ranking algorithms can be used to predict the ranks of fragment ion peaks in a peptide's experimental mass spectrum based solely on the peptide's sequence (the peak rank prediction problem). This is not a trivial task. Peptide fragmentation is a complex process that involves many competing chemical pathways. It is quite difficult to create generative probabilistic models for mass spectra that consider this wide range of chemical processes[2]. Yet generative models are not required to solve the peak rank prediction problem. The structure of this problem lends itself nicely to a ranking-based solution since the required output is an ordering of instances

[2]We note that the models described by Zhang [243, 244] are used to create a realistic theoretical mass spectrum (through a heuristic iterative process that converges at an optimal spectrum). However, they are not generative in the probabilistic sense, since they do not provide a probabilistic model describing the creation of mass spectra

Table 5.6: Peak rank prediction accuracy. The table holds statistics of the peak rank prediction accuracy for peptides of various lengths and charges. For each set of peptides, we observed how often the peak predicted to have a rank p ($p = 1 \ldots 7$) was observed in the spectrum with a rank r such that $r - 3 \leq p \leq r + 3$.

Charge	Peptide length	Rank of predicted peak						
		1	2	3	4	5	6	7
1	7	0.918	0.845	0.774	0.745	0.688	0.652	0.609
	10	0.894	0.822	0.738	0.695	0.643	0.605	0.573
	15	0.883	0.774	0.696	0.607	0.547	0.503	0.466
	20	0.779	0.740	0.679	0.584	0.426	0.429	0.429
2	7	0.957	0.916	0.905	0.870	0.820	0.777	0.704
	10	0.929	0.896	0.848	0.818	0.755	0.698	0.639
	15	0.904	0.837	0.765	0.732	0.643	0.602	0.538
	20	0.873	0.801	0.739	0.675	0.636	0.583	0.497
	25	0.857	0.774	0.735	0.668	0.605	0.553	0.497
	30	0.884	0.848	0.749	0.692	0.624	0.567	0.510
3	15	0.783	0.699	0.649	0.641	0.568	0.490	0.445
	20	0.692	0.610	0.598	0.586	0.517	0.462	0.417
	25	0.598	0.571	0.546	0.522	0.450	0.392	0.354
	30	0.601	0.498	0.500	0.478	0.431	0.368	0.328
	35	0.591	0.533	0.490	0.452	0.416	0.380	0.317
	40	0.571	0.548	0.515	0.505	0.399	0.389	0.336

rather than a portioning of the instance space (in the latter case we would consider taking a classification approach). We demonstrated how ranking-based discriminative models can be easily created using a pool of simple sequence-based features. These features get combined by the RankBoost algorithm into strong discriminative models, using a training procedure that is optimized to minimize the number of pairs of peaks on which the model makes ordering mistakes. What makes this approach feasible are the large training sets that have become available (we used approximately 320000 unique peptide-spectrum pairs), which enable us to create detailed models with minimal overfitting.

As opposed to many "black box" algorithms, such as support vector machines or neural network, the RankBoost algorithm is relatively transparent, and allows us to observe the dynamics involved in creating the models, and later, the scoring of new instances. We are able to examine the contribution of single features, and in many cases understand the logic that guided the scoring of different instances. The high-

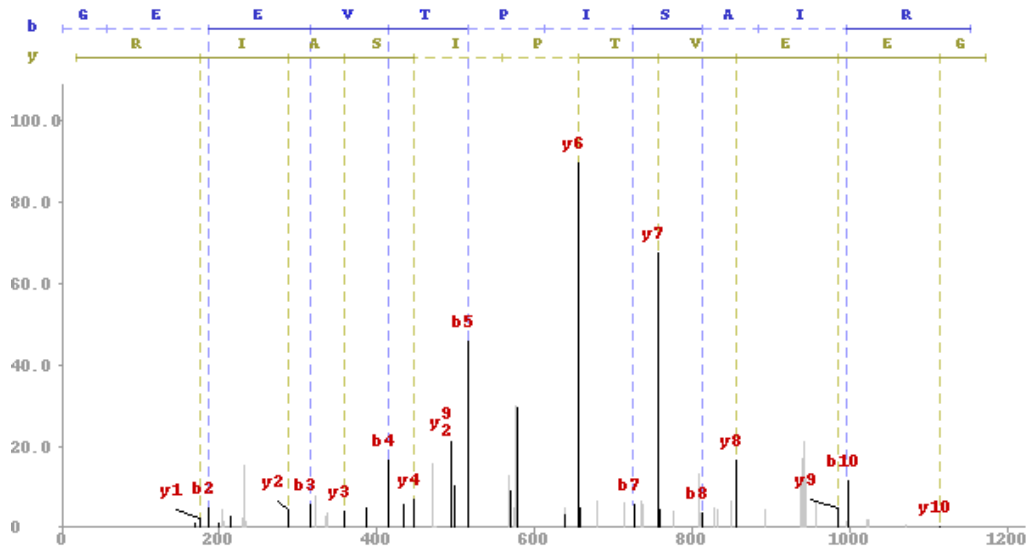
scoring features in our models reflect known chemical pathways in peptide fragmentation. Since the CID fragmentation method is well studied, we did not find evidence of novel fragmentation pathways that were not previously described in the literature. However, it is likely that given training data that was generated using a fragmentation method that has not been studied as extensively as CID, our models could supply some interesting insights into the dynamics of the peptide fragmentation.

The results in Table 5.6 indicates that our peak rank predictions are fairly accurate, and can therefore be helpful in detecting correct peptide-spectrum matches. In Chapter 6 we examine how these rank predictions can be used to enhance the performance of a novel ranking-based scoring function.

Our models' predictions for peak ranks in triply-charged peptides were slightly inferior to the predictions for singly and doubly-charged peptides. This is due to the fact that the dynamics of the fragmentation pathways in triply-charged peptides are more difficult to predict. These peptides are typically longer and contain more basic amino acids. This underscores the need for more complex features that can improve the modeling of such cases (e.g., features that describe the location of several amino acids). Another possible way to increase the models' prediction power is to incorporate into RankBoost a more powerful and expressive learning algorithm instead of the regular boosting. Alternating decision trees [78] might be good choice since they can easily account for relationships and dependencies between simple features and provide a more accurate model of the complex interactions between fragmentation pathways.

Accurate peak models become essential when trying to identify large peptides with high charge states. CID can reach charges +4 and +5, though spectra with these charges are not identified often. ETD [209] spectra routinely posses even higher charges. Top-down mass spectra [72, 117, 200], record the fragmentation of whole proteins and can posses precursor charges that reach dozens and even hundreds of protons. When such large peptides (or proteins) are fragmented, they often undergo several fragmentation events, giving rise to many internal fragments. Often, the internal fragments end up accounting for most of the spectrum's intensity. Since there is a quadratic number of possible internal ions, it becomes especially important for scoring functions to be able to predict which are the few internal fragments that are likely to be observed.

a)



b)

Peak	y_6		y_7		b_5	
Intensity	89.9		67.5		45.9	
Feature Type	Feature Value	Score	Feature Value	Score	Feature value	Score
Fragment Indicator	type $= y$	+0.43	type $= b$	+0.43	type $= y$	+0.43
Peak Location	$f^y_{max}(m) = 500$ $f^y_{rel}(m) = 0.5$	+1.32 +1.13	$f^y_{max}(m) = 400$ $f^y_{rel}(m) = 0.55$	+1.32 +1.14	$f^b_{min}(m) = 360$ $f^b_{rel}(m) = 0.35$	+0.94 +1.15
Adjacent Amino Acid*	Cut −1 is T Cut −2 is V Cut −3 is E Cut +1 is P Cut +2 is I/L	−0.29 +0.26 −0.36 +1.52 −0.22	Cut −1 is V Cut −2 is E Cut −3 is E Cut +1 is T Cut +2 is P Cut +3 is I/L	+0.79 −0.18 −0.36 +0.06 +0.57 −0.25	Cut −1 is T Cut −2 is V Cut −3 is E Cut +1 is P Cut +2 is I/L Cut +3 is I/L	−0.33 +0.26 −0.40 +1.37 −0.08 +0.03
Peptide Composition*	# N-side E = 1 # N-side G = 1 # C-side A = 1 # C-side R = 1 # C-side I/L = 1	+0.17 +0.03 −0.06 +0.36 −0.10	# N-side G = 1 # C-side A = 1 # C-side R = 1 # C-side I/L = 1	+0.03 −0.06 +0.36 −0.10	# N-side E = 1 # N-side G = 1 # C-side A = 1 # C-side R = 1 # C-side I/L = 1	+0.17 −0.03 +0.03 +0.12 −0.04
Terminal Amino Acids*	N-term aa is G C-term aa is R	−0.02 +0.50	N-term aa is G C-term aa is R	−0.02 +0.50	N-term aa is G C-term aa is R	+0.11 −0.50
Total Score		4.67		4.23		3.23

Figure 5.6: Example of computation of peak rank scores for the three strongest peaks in the spectrum of the peptide GEEVTPLSALR. Part *a* displays the experimental spectrum of the peptide GEEVTPLSALR. Part *b* lists the feature vectors used to compute the rank scores for the three strongest peaks in the spectrum: y_6, y_7, and b_5.(*) Different adjacent amino acid feature, peptide composition features and terminal amino acid features are created for each fragment type (b, y). This explains why the scores for the same type of feature (e.g., Cut −1 is T) are different between y_6 and b_5.

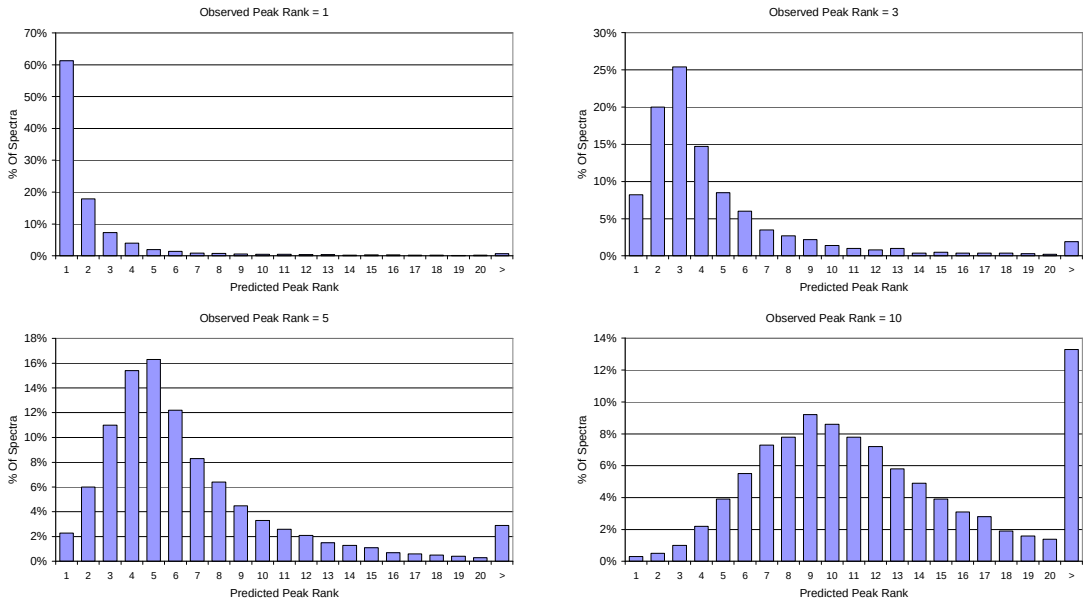


Figure 5.7: Peak Rank Prediction Histograms. The figure shows histograms of the predicted peak ranks for peaks observed in the experimental spectra with ranks 1,3,5 and 10. The results were collected from 5000 doubly charged peptides of length 10. Each observed peak was assigned a predicted rank between 1 and 36 (9 internal cleavage sites $\times$ 4 fragment types).

6

Scoring Peptide-Spectrum Matches

6.A Introduction

Scoring functions are one of the key components of any algorithm used to identify peptides from experimental mass spectra. Scoring functions assign a numerical value to a peptide-spectrum pair (P, S) expressing the likelihood that the fragmentation of a peptide with sequence P is recorded in the experimental spectrum S. The problem of scoring peptide-spectrum matches has received considerable interest in the community over the years [8, 25, 35, 38, 39, 49, 59, 69, 73, 93, 136, 146, 162, 181, 219, 229]. Most scoring functions involve creating a *generative* statistical model for $Prob(S|P)$, the probability that spectrum S was created from a fragmentation of peptide P, and then selecting $P^* = \text{argmax}_P Prob(S|P)$ as the likeliest peptide that created S. The hope is, that if the model $Prob(S|P)$ is sufficiently accurate, then the peptide that maximizes $Prob(S|P)$ will indeed be the true peptide recorded in S. The probability $Prob(S|P)$ is often converted to a score by using a log ratio test that compares $Prob(S|P)$ to the probability obtained from a simple null hypothesis, such as a model that assumes that all peaks are distributed randomly [25, 35, 49, 73, 93, 146].

The problem with using generative approaches to scoring is that peptide fragmentation is an extremely complex process, that is not easily represented with high

fidelity by simple statistical models. The current scoring functions are usually sufficient when the search space is small, such as searching against a small set of protein sequences. However, when we increase the search space, by searching against a large database such as the six-frame translation of the human genome, or by performing de novo sequencing (which effectively searches the space of all peptides), the number good identifications typically decreases significantly. In these large search spaces, generative scoring models often lack the sufficient power to discriminate between the correct peptide-spectrum matches and the many close false ones (like the homeometric peptides discussed in Section 4.B.1).

Another characteristic of many scoring functions, especially those used for de novo sequencing [49, 69, 73, 146], is that they are *additive*, i.e., the final score for a peptide-spectrum match can be decomposed into contributions made by the individual amino acids in the solution. Though this fact is appealing from an algorithmic perspective, making it easy to use methods like dynamic programming to efficiently find the optimal solution, it does tend to limit the type of features that can be used in the scoring models. In particular, it makes it difficult to look at global features that pertain to the entire peptide sequence (such as the distribution of basic amino acids, presence or absence of fragment ions of a certain type, etc.). Such features can play an important role in discriminating between very close peptide-spectrum matches that are typically encountered in de novo sequencing.

Scoring peptide-spectrum matches is at times viewed as a classification task, in which we create a model that is trained to separate between the class of all instances of correct spectrum-peptide matches and the class of incorrect instances. We argue below that scoring, especially in the context of de novo sequencing, should be viewed as a ranking task in which the goal is to bring the correct peptide to the head of a spectrum's candidate peptides list. By looking at the scoring problem in this different light we are able to use a different set of tools – namely, discriminative ranking models – to solve the problem, instead of the more conventional generative model approach. As we shall see below, this can significantly improve the performance of our algorithms.

6.A.1 Scoring De Novo vs. Scoring Database Search Results

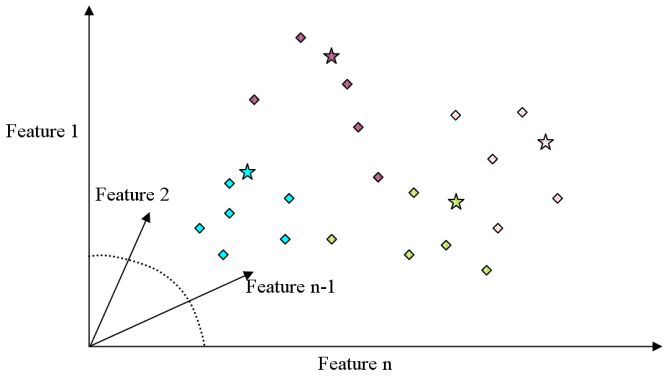
There is a fundamental difference between the way scoring functions are used with de novo sequencing and the way they are used in database searches. With de novo sequencing, we search the space of all peptides, and we assume that the correct peptide sequence is in our search space[1]. Therefore, the purpose of the scoring function is simple: For each spectrum, bring the correct peptide to the top of the list of candidate peptides. However, when scoring database search results, the score has a dual purpose. Like in the case of scoring de novo sequencing results, the scoring function still needs to bring the correct candidate to top of the spectrum's candidate peptides list. But there is another stronger constraint. The score needs to (ideally) assign *all* correct peptide-spectrum matches a higher score than *all* incorrect peptide-spectrum matches. This more stringent scoring goal is needed because when we search against a database, we cannot assume that the correct peptide is necessarily in the database. In fact in a large proportion of the cases, the database does not contain the correct peptide for a query spectrum (e.g., the peptide might originate from an unknown gene). Even when searching a six-frame translation of a genome, our peptide might be the result of alternative splicing, contain a mutation, or have a modification that was not entered in the search options, and therefore would not be included in the peptide search space. If we would accept all top scoring peptides as being correct identifications, our results would contain many false positives.[2]

Figure 6.1 illustrates the difference between the goal of scoring in de novo sequencing, and the more stringent goal of scoring database search results. A star marks the location of a correct peptide-spectrum match, and a diamond marks the location of an incorrect peptide-spectrum match. The figure contains peptide-spectrum matches for 4 spectra, each designated by a different color. In part *a* of the figure we see the matches as points in an $n-$dimensional feature space, prior to applying scoring. The scoring

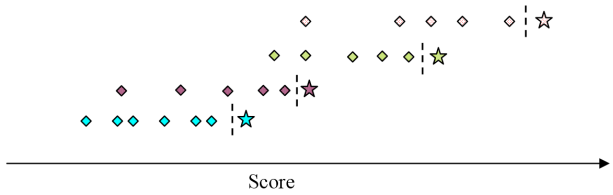
[1] With this statement we make a couple simplifying assumptions. When performing de novo sequencing, we usually ignore the fact that a peptide might contain a modification. It is often too difficult to perform de novo sequencing in such cases without knowledge of the peptide's specific modification. In addition, we ignore the fact that a spectrum might not contain peptide information at all, since in such cases de novo algorithms do not return any reasonable scoring sequences.

[2] Using a decoy database in the search helps to control the number of such false positives that can be expected in the search results [10, 60, 97].

a) Peptide-spectrum matches before scoring



b) After scoring for de novo sequencing results



c) After Scoring for database search results

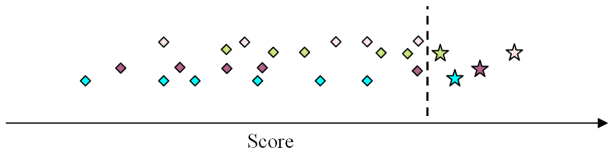


Figure 6.1: Different goals of scoring functions for peptide-spectrum matches. Correct peptide-spectrum matches are marked with a star, and incorrect peptide-spectrum matches are marked with a diamond. The illustrations above show peptide-spectrum matches for 4 different spectra, which are marked using four colors. Part *a*, on the top, shows the peptide-spectrum matches as points in an $n-$dimensional feature space, prior to applying scoring. Part *b* shows the result of a scoring function designed for de novo sequencing, whose goal is to bring each star ahead of the diamonds of the *same* color. Part *c*, shows the results of a scoring function designed for a database search, where the goal is to bring *all* stars ahead of *all* diamonds.

function maps each peptide-spectrum match to a real number. In part *b* we see scoring results that are sufficient for the assumptions made when performing de novo sequencing. For each set of peptide-spectrum matches with the same color, the star (correct match)

has a higher value than the diamonds (incorrect matches). In this case, we do not care if diamonds from one color have a higher value than a star from another color, since in de novo sequencing, we do not compare between matches from different spectra. In part *c* we see scoring results that accommodate the more stringent requirements of database search scoring. This score brings all correct peptide-spectrum matches (stars) ahead of all incorrect peptide-spectrum matches (diamonds).

Database scoring functions can usually handle the more stringent requirements of them because their search space is typically small, and they are not likely to be confronted by too many high-scoring incorrect peptide-spectrum matches. In addition, database scores often look at additional factors, namely the difference between the top and second best scoring matches when they decide wether or not to accept an identification (e.g., the ΔC_n measure in Sequest's score [61]). Thus, in some cases, algorithms can still accept low-scoring identifications, if the gap between the first and second best identifications is large enough. However, the circumstances change when the size of the search space increases substantially, such as when searching a six-frame translation of a genome. Under these conditions, there are many more strong but incorrect peptide-spectrum matches, and the database scoring function's performance deteriorates significantly (see in Section 6.C.5). A more discriminating scoring function becomes essential in these circumstances.

The difference between the goals of scoring functions for de novo and database searches can influence the choice of the computational tool we end up using. There is great diversity in the experimental mass spectra that are obtained from the fragmentation of different peptides; some peptides generate full fragmentation ladders where most of the *b*- and *y*-ions can be observed, while others fragment poorly. Consequently, when the peptide spectrum-matches are mapped to a feature space, there is a poor separation between the classes of correct and incorrect peptide-spectrum matches. In Figure 6.1*a*, we see that the stars (correct peptide-spectrum matches) and the diamonds (incorrect peptide-spectrum matches) are quite overlapping. Furthermore, diamonds from one color are much closer to the star of their own color than they are to diamonds of other colors. This phenomenon is especially widespread when the peptides are results of de novo sequencing, where one can transform a correct match to an incorrect one by a trivial

action like inverting two amino acids in the peptide sequence (which would hardly affect most feature values). Due to this poor separation, if we chose a score that is designed to separate all the stars from all of the diamonds (as illustrated in Figure 6.1*c*) – which is essentially a classification task – it will have its work set out for it. However, if we relax the demands of the scoring function, requiring it only separate between instances from the same spectrum (as illustrated in Figure 6.1*b*), this becomes a ranking task. In particular, as we shall see below, the problem becomes amenable to a solution using discriminative ranking algorithms.

6.B A Discriminative Scoring Model For Peptide-Spectrum Matches

In this chapter we develop a new scoring function that is more data-driven than previous approaches. Our goal is not to create an accurate *generative* model $Prob(S|P)$, which is inarguably a difficult task. Instead, we desire a scoring algorithm that performs well on a simpler *discriminative* ranking task: Given a spectrum S and a set of candidate peptides $P_1, \ldots, P_k$, we want the model to be able to assign scores to the peptides according to how well their expected fragmentation pattern matches the observed spectrum S. Our goal being only that the correct peptide receive the highest score – we do not concern ourselves with the margin between the correct peptide's score and the scores assigned to the other incorrect peptides.

The most important component of our scoring models are the feature functions they use. Our models draw on a diverse set of features, created using domain knowledge, that each in there own way reflect different characteristics that can help distinguish between correct and incorrect peptide-spectrum matches. In total, the models can contain up to 225 features (though not all get selected in each model). We grouped these features into different classes, as described below. For each feature class, we give examples of the most prominent features (the ones that most influence the ranking score). For many of the features described below we also provide figures with the feature score plots. The values for the plots were taken from the model trained for scoring de novo sequences generated from doubly-charged spectra with parent mass 1100-1300 Da (about 9-15 amino

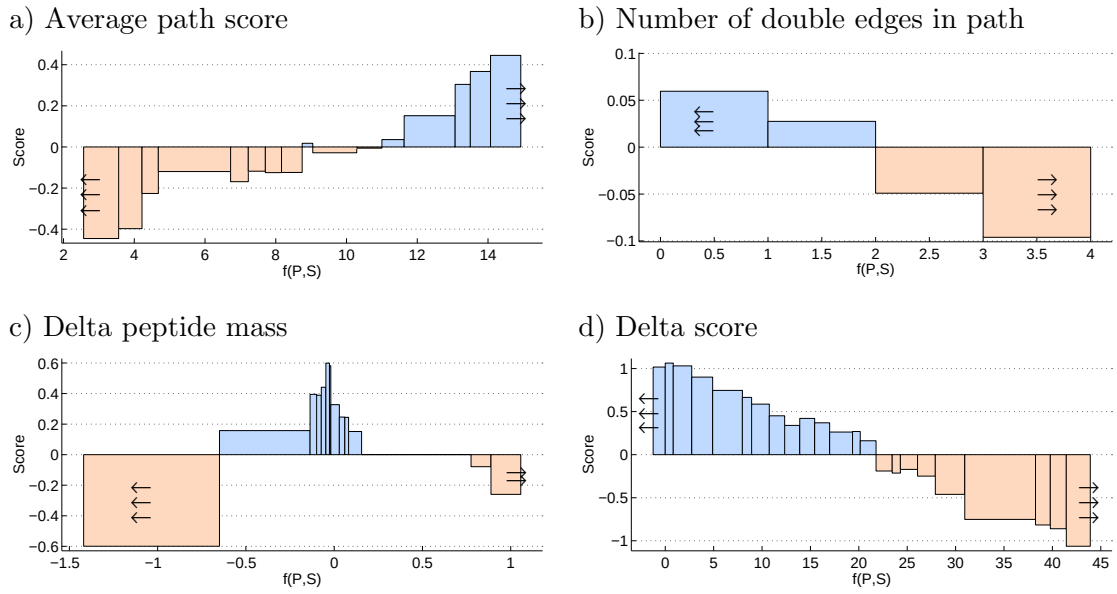


Figure 6.2: Spectrum graph features. The x-axis holds feature function values $f(P, S)$ that are computed when matching a peptide P to a spectrum S. The y-axis gives the score assigned by the model to the different feature function values.

acids). The feature functions described below are also used later for other scoring tasks besides reranking de novo results, such as scoring tags, and scoring database search results.

Spectrum Graph Features

The spectrum graph scoring function is based on detailed probabilistic models (see Chapters 2 and 4); it considers important factors such as dependencies between fragment ions, the observed peak intensities, the influence of flanking amino acids, and the location of the cleavage site in the peptide. An incorrect peptide-spectrum match is likely to contain more cleavage sites that are either missing detected fragments altogether, or have combinations of observed ions that are less likely (and are scored poorly in the spectrum graph). To capture this information, we examine several aspects of the spectrum graph scores that can be informative. Since we use the same model to compare peptides that can have different lengths, we cannot use a fixed set of features that is length dependant (e.g., score at cleavage 1, score at cleavage 2, etc.), rather we use features that are invariant to length such as the total score, or average cleavage score. Below is a list of some of the prominent spectrum graph-based features:

- **Total and average path score [6.2a]** - The score of a peptide's path in the spectrum graph is computed using a likelihood ratio score (see Section 2.B). On average we can expect the path of the correct match to be higher than the path of an incorrect peptide-spectrum match. To avoid biases that are due to the predicted peptide's length, such as when comparing a partial peptide prediction to a full one, it is also beneficial to look at the average path score (total score divided by the number of amino acids in the predicted peptide). Figure 6.2a depicts the plot of the average path score feature function. It shows a monotonically increasing reward for having a high average path score.

- **Minimal cleavage scores** - Usually the top scoring de novo sequences are quite similar to the correct sequence, but make suboptimal short "detours" in the spectrum graph. In such cases they are likely to score lower at certain cleavage sites. An informative feature can be to look at the minimal (and second and third lowest) scores assigned to cleavage sites in the peptide's path.

- **Number of double edges used in the path [6.2b]** - Most peptide bond cleavages produce fragment ions that are detected as peaks in the spectrum. However, there are often cases where a bonds do not produce detectable peaks in the spectrum, and therefore our spectrum graphs contain double edges. However, excessive use of double edges in a path usually indicates that the path belongs to an incorrect peptide. In Figure 6.2b we see the scores assigned due to the presence of different number of double edges. Having no double edges receives a premium of +0.05 to the rank score, while having 3 or more double edges reduces the score by 0.1.

- **Number of forbidden node pairs** - Forbidden node pairs occur when a single peak is assigned to more than one fragment (e.g., it is considered to be both a b_8 and a y_4). If one or more such a cases are detected, the score for the peptide-spectrum match receives a penalty of -0.5.

- **Delta of peptide mass [6.2c]** - The spectrum graph is constructed while allowing a certain error tolerance for peak masses (typically we used 0.5 Da.). Such mass errors can accumulate as we traverse along the peptide's path. However with

correct peptides, the typical difference between the sum of the mass of the peptide's amino acids and the mass of the path is not great (the mass of a path is defined as the mass of the last node minus the mass of the first node). Figure 6.2*c* shows that while having a delta mass near 0 yields a premium of +0.6, having a negative delta mass beyond 0.65 Da is not common for correct peptide-spectrum matches, and thus brings a large penalty of −0.6

- **Delta rank** - When scoring de novo sequences, we are given a list candidate peptides that can be ranked according to their paths' scores. Since often the highest scoring de novo paths belong to correct peptides, the original path ranks are obviously helpful when reranking de novo results.

- **Delta Score [6.2d]** - Often the de novo search can generate many high-scoring similar de novo paths that differ from each other by only one or two amino acids. In such cases, the correct peptide might have a relatively low rank, but its score will not be much lower than score of the highest ranked peptide. It is therefore useful to have a feature that relies on the difference in score, rather than the difference in rank (as does the "Delta rank" feature mentioned above). Figure 6.2*d* shows that being close to the optimal score is a characteristic of many correct peptide-spectrum matches. There is a monotonically decreasing premium that is approximately +1 when the path score is up to 3 away from the optimum, which turns into a penalty once the score difference exceeds 21.

The relatively high weight assigned to the "Delta Score" feature indicates the importance of the original ranking of the de novo results (according to PepNovo's score). In essence, PepNovo's output is ordered solely according to this feature. All the other features described in this section serve to refine the ordering induced by this feature, and increase the number of cases in which correct lower-scoring peptides are ranked above incorrect higher-scoring solutions.

Peak Rank Prediction Features

In Chapter 5 we described an algorithm for predicting the expected relative ranks a peptide's fragment ion peaks. Though this is a step short of predicting the

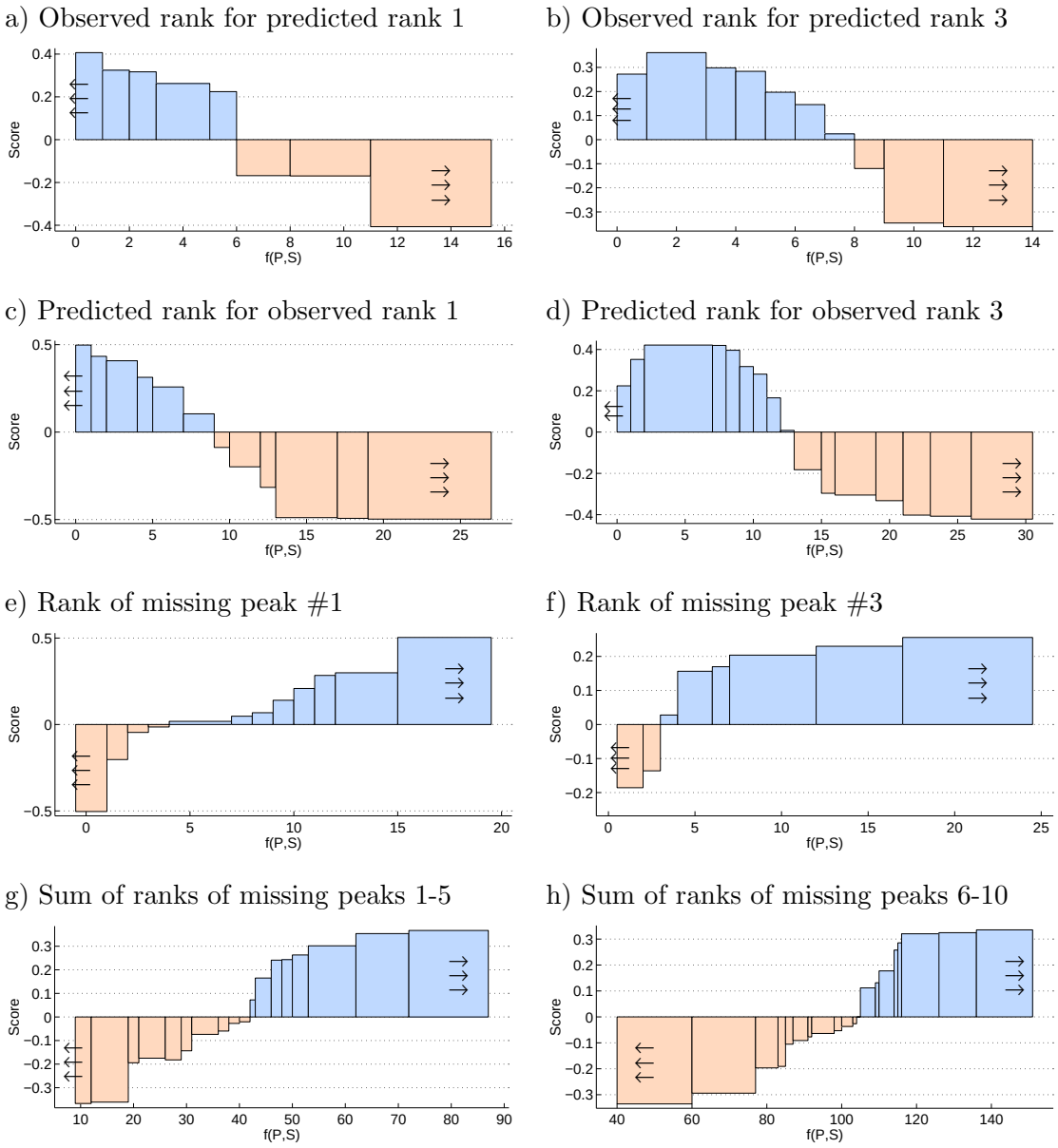


Figure 6.3: Peak rank prediction features. The x-axis holds feature function values $f(P, S)$ that are computed when matching a peptide P to a spectrum S. The y-axis gives the score assigned by the model to the different feature function values.

actual theoretical spectrum [243, 244], the predicted peak ranks still carry a lot of information that can be used to improve the scoring of peptide-spectrum matches. Since our prediction of peak ranks is most accurate for the highest ranks, the features mostly focus on these ranks. Some of the most useful peak rank prediction features include:

- **Observed rank for peak with predicted rank X, ($\mathbf{X = 1, \ldots, 7}$) [6.3a,6.3b]**

- This type of feature examines the difference between the ranks observed for peaks vs. the ranks we predicted for them. For each predicted fragment peak with rank X, the feature function reports the actual rank observed for that fragment ion's peak in the experimental mass spectrum (a rank of ∞ is given if the peak is not observed in the spectrum). Figures 6.3a and b depict the scores assigned to the features that examine the peaks with predicted ranks 1 and 3, respectively. The features give a premium if the observed rank is close to the feature's predicted rank. This premium decreases as the observed rank gets farther from the predicted one. In both cases if the observed rank is above 12, it is treated the same as case of a predicted peak being unobserved in the experimental spectrum.

- **Predicted rank of peak with observed rank X, ($\mathbf{X = 1, \ldots, 7}$) [6.3c,6.3d]** - This type of feature uses peak rank predictions the other way around, and examines what is the rank that was predicted by the model for the peak observed in the spectrum with rank X. Figures 6.3c and d depict the scores assigned to the features that examine the peaks with observed ranks 1 and 3, respectively.

- **Rank of missing peak #X, ($\mathbf{X = 1, \ldots, 10}$) [6.3e,6.3f]** - This feature examines the theoretical masses of fragment peaks, according to the order of their predicted ranks (starting with the peak predicted to have the highest intensity). The feature notes the rank of the X'th missing peak (i.e., there was no peak detected in the spectrum at the expected mass). Figures 6.3e and f depict the features that examine the first and third missing ranks, respectively. The models assign penalties when the ranks of the missing peaks are high (since this indicates a poor fit between the predicted ranks and the observed spectrum).

- **Sum of ranks of missing peaks 1-5,6-10 [6.3g,6.3h]** - This type of feature is more general than looking at each rank X individually, since it carries information on the occurrence of multiple missing peaks (which is a strong indication that the peptide is incorrect). Figures 6.3g and h depict the features that examine the sum of missing ranks 1-5 and 6-10, respectively.

Peak annotation features

The spectrum graph features mostly evaluate combinations of fragments that involve specific cleavage sites. However, it is also beneficial to take a global look at how well the peptide explains the spectrum's peaks. This type of information is not easily conveyed when additive score functions are used. Some of the most useful features that capture this global information are:

- **# Annotated peaks in top 25,50 peaks [6.4a]** - A correct peptide should typically explain many of the strongest peaks in the spectrum. Figure 6.4*a* depicts the scores assigned by the model to this feature. A good match tends to explain a large proportion of the top 25 peaks.

- **% Explained intensity** - This feature measures how much of the total intensity in the spectrum can be assigned to the peptide's fragment ions. Generally, we expect a good match to explain a large proportion of the experimental spectrum's intensity.

- **# of peak annotations for fragment $\mathbf{X = b, y, a, y^{+2}, y - H_2O, \ldots}$ [6.4b]** - Correct peptides are likely to explain many types of fragments. Since the spectrum graph score looks at individual cleavage sites, it cannot detect events that are probable for any single cleavage site, but less probable for a whole peptide. For example, even though with doubly charged tryptic peptides, the probability of observing a y^{+2}-ion at any given cleavage is less than 50%, it is quite unlikely not to detect any y^{+2}-ions at all. Figure 6.4*b* shows that such cases are penalized by subtracting 0.65 from their scores. However, peptides for which we find 3 or more y^{+2}-ion peaks receive a large premium.

- **% of peak annotations for fragment $\mathbf{X = b, y, a, y^{+2}, y - H_2O, \ldots}$ [6.4c,6.4d]** - This feature is similar to the feature above, however gives values that are normalized according to the peptide's length. Figures 6.4*c* and *d* depict the scores given to the features measuring the proportion of annotated *b*- and *y*-ions, respectively.

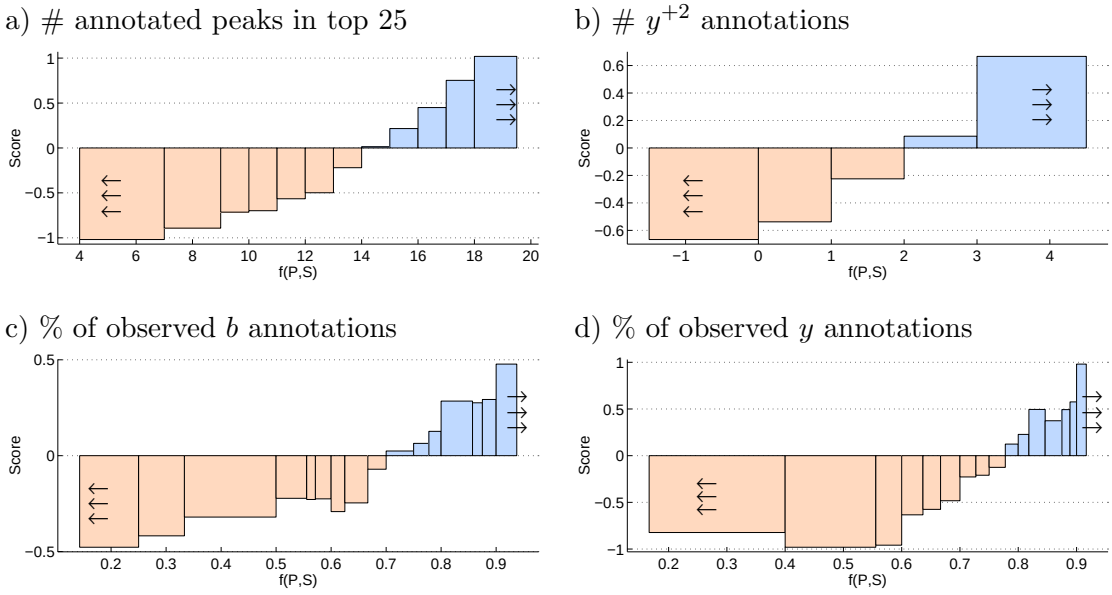


Figure 6.4: Peak annotation features. The x-axis holds feature function values $f(P, S)$ that are computed when matching a peptide P to a spectrum S. The y-axis gives the score assigned by the model to the different feature function values.

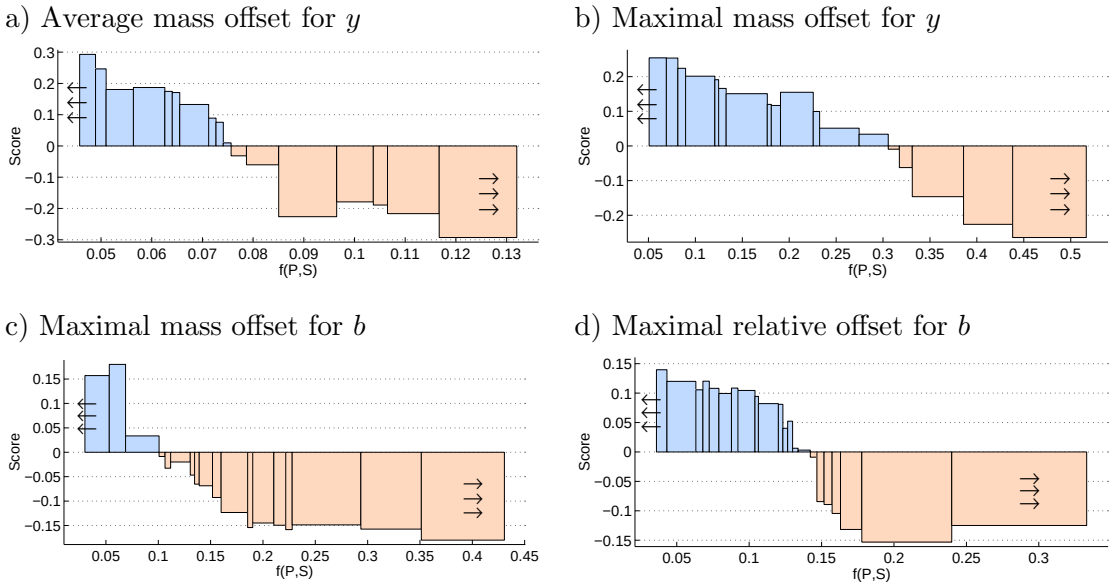


Figure 6.5: Peak offset features. The x-axis holds feature function values $f(P, S)$ that are computed when matching a peptide P to a spectrum S. The y-axis gives the score assigned by the model to the different feature function values.

Peak offset features

When annotating fragment ions, we generally tolerate a mass differences of up to 0.5 Da between the expected mass of a fragment (as computed from the peptide

sequence) and the actual mass observed in the spectrum. However, most of the true fragment peaks observed in spectra are much closer to their expected mass, usually being less than 0.1 Da away. A peptide that has many fragment peaks with a relatively large offset from their expected mass is likely to be relying on spurious opportunistic peak matches, and is therefore more likely to be incorrect. This type of peak offset information is most useful with the most abundant fragment ions, which are b, y, so offset related features focus only on them:

- **Average mass offset for fragment b/y [6.5a]** - This feature looks at the average mass offset of all identified b (or y) peaks. Figure 6.5a depicts the scores assigned by the model to the average offset measured for the peptide's y-ion fragments. Typical correct peptide-spectrum matches have an average peak offset of less than 0.085 Da; larger offsets are penalized.

- **Maximal mass offset for fragment b/y [6.5b,6.5c]** - Often a bad peptide-spectrum match contains an opportunistic use of a single peak (this is especially true in de novo sequencing). Many times such peaks are not close to the expected mass. Looking at the maximal offset observed for a fragment, rather than the average, can be more discriminating in these cases. Figures 6.5b and c depict the scores given to the maximal offset features for y- and b-ions, respectively.

- **Maximal relative offset for fragments b/y [6.5d]** - Sometimes spectra contain systematic biases in the peak locations (e.g., there is a fixed offset to most of the peak masses or an offset that increases with peak mass). In such cases the absolute peak offset might be relatively high, but we still can detect good peak matches by examining the mass of *successive* fragment ions. For example the offset of two successive b separated by amino acid A is computed as $b_n - b_{n-1} - mass(A)$. Figure 6.5d depicts the score assigned by the model to the feature examining the maximal relative offset of b-ions.

Sequence Composition Features

Proteins are not random sequences of amino acids. They often contain conserved, or characteristic patterns that are responsible for inducing a specific spacial

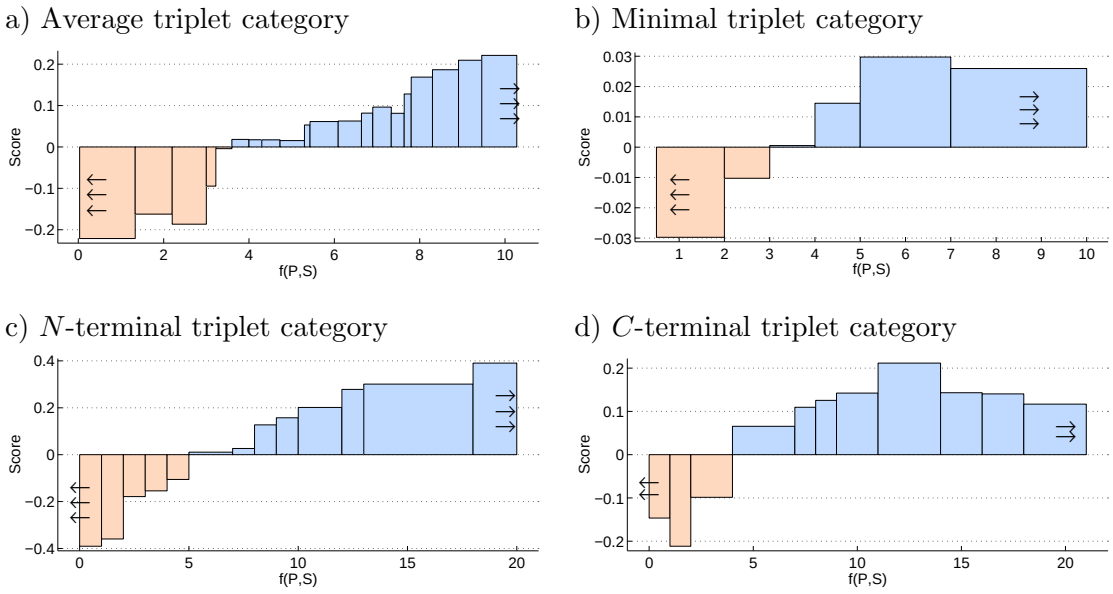


Figure 6.6: Sequence composition features. The x-axis holds the feature values $f(P, S)$ that are computed when matching a peptide P to a spectrum S. The y-axis gives the score assigned by the model for the different feature values. Triplets of amino acids were assigned to categories according to their frequency is proteotypic peptides. The categories range from 1, for the most rare triplets, to 20 for the most frequent.

conformation or for providing certain function. In addition, certain amino acid patterns are more likely to be ionized and detected using MS/MS than others (e.g., basic amino acids are usually required for effective peptide ionization). These observations gave rise to the notion of proteotypic peptides [45, 138, 215], peptides that are most likely to be confidently identified by MS/MS methods. Maintaining a list of proteotypic peptides is of course not suitable for an unrestricted general-purpose scoring function. However, many of the characteristics of proteotypic peptides can be captured using simple features that pertain to the peptide's amino acid composition.

We focused our efforts on amino acid triplets. These are relatively short sequences, and thus could not be trained to fit specific peptide sequences. We examined a large set of proteotypic peptide sequences [45], and computed frequency statistics for all possible amino acid triplets. We then divided the triplets into 20 categories according to their frequency. Category 1 contains the least frequent amino acid triplets (e.g., WKW, KCR, YRM), and category 20 contained the most frequent triplets (e.g., GGG, ELL, ALA). Similar tables were constructed for the first triplet (the first three amino

acids on the N-terminal side) and the last triplet (the last three amino acids on the C-terminal sides), which can have different frequencies due to the specificities of the enzymatic cleavage. Features derived from these tables included:

- **Average/minimal triplet category [6.6a,6.6b]** - For a peptide of length n, there are $n-2$ triplets for which we compute the average and minimum triplet category values.
- **Category of triplets on N-/C-terminal sides [6.6c,6.6d].**

We also examined the composition by creating features of the type

$$f_{\#X}(P) = \#\text{ of amino acids X in the peptide P.}$$

Using such simple features helped correct biases in the regular de novo sequencing scoring. For instance, if a peptide had one glutamine there was a score penalty of -0.4 and if there were two or more glutamines, there was a score penalty of -0.7. The reason this amino acid received these penalties is that glutamine (Q) has the same mass as alanine (A) + glycine (G), and it often wrongfully replaces these two amino acids in the de novo sequencing results. Likewise, if the peptide contains tryptophan (W), there is a penalty -0.31. W has the same mass as A+D,E+G, and V+S, so it too is likely to cause sequencing mistakes).

Summary

Our rank models typically contain most of the 225 scoring features, but not all feature functions carry the same weight. As depicted in Figures 6.2-6.6, some feature functions are assigned high scores, even reaching ± 1, while other feature functions are assigned much lower scores. All features are important for optimal ranking (otherwise they would not have been included in the model). It is true that a small set of features that posses high scores can perform most of the coarse ranking process; moving the good peptide-spectrum matches up in the ranks and the bad matches down. However, for close calls, such as correctly ranking very similar peptides obtained by de novo sequencing, the models rely on the many other features that have small score values (e.g., peak offset

features, composition features, etc.), to perform the fine tuning and give the correct peptides a slightly higher score, which is sufficient to push it ahead to the top of the list.

6.C Experimental Results

We now turn to examine how our new scoring model can be used to improve the results of de novo sequencing, tag generation and database searches. We first describe the process involved in training rank score models for de novo sequences.

6.C.1 Model Training (for de Novo reranking)

We used the partitioning of the training data into 13 sets according to charge and parent mass as described in Table 5.1, and trained a score model for each partition separately. Each partition contained 13000-45000 spectra which were used to create 400000 training samples as follows. We performed de novo sequencing using PepNovo[3] on each training spectrum S_i, and retrieved the top-scoring 2000 sequences that were at least 6 amino acids long. If none of de novo sequences was correct, we excluded the spectrum from the training set. From the set of 2000 de novo sequences we selected the highest scoring correct peptide sequence and used it as the positive sample P_i^+ (the highest scoring correct sequence was usually also the longest correct sequence). From the remaining 2000 incorrect de novo sequences we randomly sampled $k = 10 - 30$ sequences and used them as negative samples $P_i^{-1}, \ldots P_i^{-k}$. Note that we did not sample the sequences uniformly from ranks 1-2000, rather, we gave more weight to higher ranking sequences which typically are responsible for most of the ranking errors.

Using the feature functions described in Section 6.B, we created a point x_i^+ in the feature space to represent the peptide-spectrum match of P_i^+ and S_i, and similarly created points $x_i^{-1}, \ldots, x_i^{-k}$ to represent the peptide-spectrum matches of the incorrect de novo sequences. The points x_i^+ and $x_i^{-1}, \ldots, x_i^{-k}$ where used to create a set of k ordered pairs $\{(x_i^{-i}, x_i^+) | i = 1, \ldots, k\}$, which were added to the model's feedback function Φ (see Section 5.A.2). 70% of the spectra's sets of samples were randomly selected to participate

[3]The version of PepNovo used for these experiments is the newer implementation of the algorithm described in Section 4.B.2. We used the large training sets to create specific scoring models for each partition according to charge/size, as described in Table 5.1.

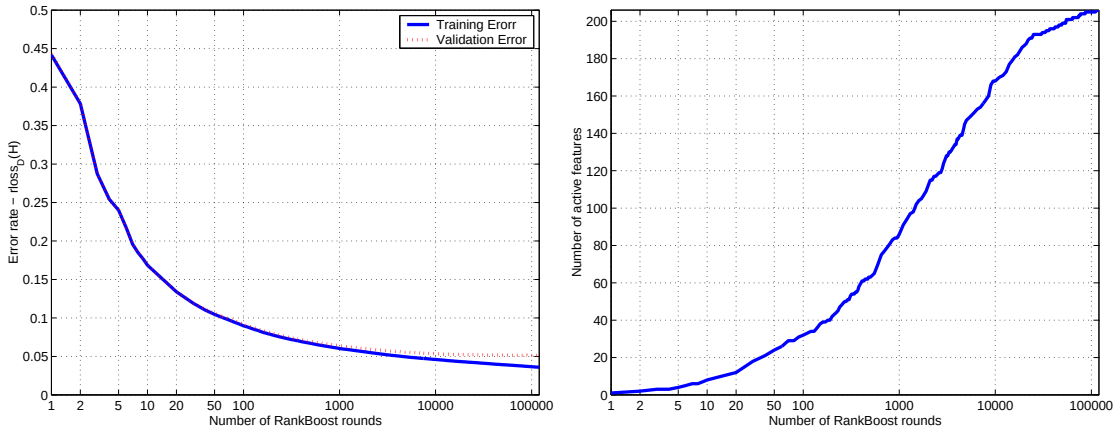


Figure 6.7: Training of de novo score model. The graph on the left displays the training and validation error rates after running the RankBoost algorithm for various numbers of rounds. The graph on the right displays the number of active features in the model (i.e., features that have a nonzero wight). The x-axis displays the number of boosting rounds using a logarithmic scale. The figures were generated for a training set of doubly-charged peptides of mass 1100-1300.

in the training of the model, while the remaining 30% served as a validation set which was used to determine when to terminate the model's training (to avoid overfitting).

Training each score model typically required less than 100 CPU hours. The score models typically converged after less than 100000 rounds. Figure 6.7 depicts the progression of the training of the model for doubly charged peptides with parent mass 1100-1300 Da. The left side of the figure shows the training and validation errors. Most of the ranking error is eliminated within a very small number of rounds (the error decreases from 45% to 10% within 50 rounds). The figure also shows that most of the error reduction is done using a small set of features (the graph on the right shows that approximately 25 features are required to achieve this error reduction). Continuing to 10000 rounds lowers the validation error to 5.35%, which is only 0.22% higher than lowest validation error 5.13%, that is obtained after 90000 rounds. Therefore, by not seeking to fully optimize the model, we can save considerable time with the training. The graph on the left also shows that as the training progresses overfitting starts to become a problem (this is evident from the widening gap between the training error and validation error that does not decrease in the same pace). However, despite the overfitting, the overall validation error kept on decreasing, and we ended up choosing the model configuration

that had the lowest validation error.

6.C.2 Benchmark Results for De Novo Sequencing

De novo sequencing of low-resolution MS/MS data is a difficult task. It is unreasonable to expect high accuracy rates from single de novo predictions, since often there are many similarly high-scoring candidates to choose from. Furthermore, the most commonly used applications of de novo sequencing, such as database filtration using tags [76, 139, 197, 208, 212, 219] or homology-based database searches [89, 192, 194, 233], can easily use multiple de novo predictions instead of a single one. It is in these circumstances that the advantage of ranking comes into play. Not only does our score increase the accuracy of the top predicted sequence, but it also significantly increases the chances of having a correct sequence in a small set of candidates.

We conducted several benchmark experiments to test the performance of our new scoring function in the context of de novo sequencing. We used several test datasets, including two test sets that were previously used in the literature:

- OPD280 - The set of 280 spectra of doubly charged spectra used to benchmark PepNovo in Section 2.C, and also used in refs [69, 73, 146].
- ISB769 - A set of 769 spectra from the ISB dataset [119], which was used in ref [146].
- HEK8, HEK10, HEK12 - 3 Sets of 1000 doubly charged spectra that were selected from the HEK293 dataset [218]. Each set contained spectra of peptides of specific lengths: 8,10, and 12 amino acids, respectively.

Previous de novo sequencing benchmarks experiments mostly focused on predicting a single sequence per spectrum [69, 73, 146, 167]. In these cases it made sense to look at the precision (ratio of correct amino acids in the predictions). However, when predicting multiple sequences, this notion is not well defined. Instead, we examine the proportion of test spectra for which one of the de novo predictions is completely correct, and also look at the rank in which a correct prediction first occurs.

Most publicly available de novo sequencing algorithms typically return a single sequence prediction. The newly developed MS-Dictionary algorithm of Kim et al. [120]

takes a novel approach of combining de novo sequences and a database search. It uses dynamic programming and probabilistic scoring to generate large ranked lists (dictionaries) of possible peptides for query spectra. These peptides are then compared to a database via pattern matching (similar to the approach we explored in Chapter 4). We examined MS-Dictionary's results with two settings, one that assumes that the peptides are tryptic (and thus lists only peptides that end with K or R), and the other that makes no assumption about which digestion enzyme was used. In addition, we ran experiments with the Peaks [136] de novo sequencing algorithm (PEAKS Online 2.0) which is one of the best commercial de novo sequencing algorithms available. The Peaks de novo algorithm only outputs 5 sequences per query spectrum.

Our experiments proceeded as follows. For each spectrum tested, we ran PepNovo and generated the top 2000 scoring sequences. We then reranked the sequences using the our new scoring function described above. In the results below we compare between the algorithm's performance with and without the reranking stage. On average the running time required per spectrum was 1-2 seconds, depending on the peptide's length. This running time usually divided equally between the de novo sequencing and the reranking. When the true peptide sequence was short (8-10 amino acids long), PepNovo typically predicted the entire sequence. However, with longer peptides, whose spectra are often incomplete and lacking detected fragment ion peaks for the amino acids near the terminals, PepNovo sometimes only predicted partial sequences (akin to long sequence tags).

Figure 6.8 shows benchmark results of the algorithms on the OPD280 and ISB769 datasets. In both datasets we see that PepNovo has significantly higher rates of correct predictions, especially when we look at a small set of de novo solutions. PepNovo uses a much more sophisticated scoring scheme than MS-Dictionary, which explains the large performance gap when small sets of predictions are concerned. In addition, MS-Dictionary is designed to predict only complete de novo sequences. Both the OPD280 and ISB769 datasets include some sequences longer than 14 amino acids (the length limit for which MS-Dictionary is deemed effective), which also explains the lower performance of this algorithm on these datasets. The Peaks algorithm displays accuracy rates that are slightly lower than PepNovo's without ranking.

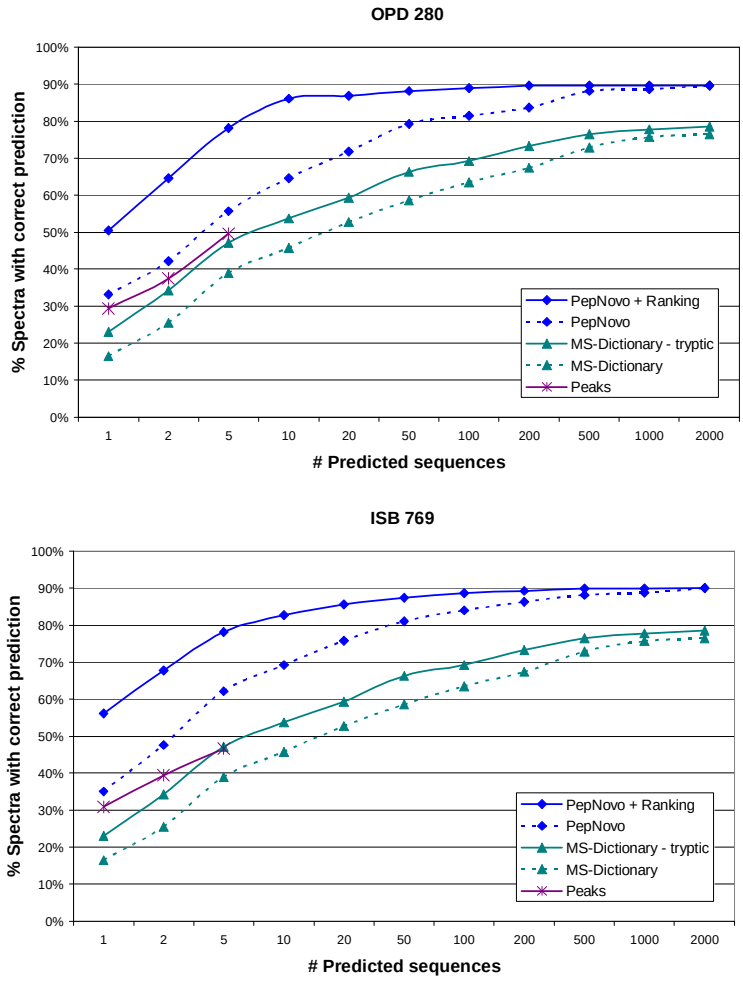


Figure 6.8: Benchmark results for OPD280 and ISB769. The plots show results for PepNovo (with and without reranking), MS-Dictionary (with tryptic only and non-restricted predictions), and Peaks. In each plot the x-axis shows the size of the set of highest scoring predicted sequences (1-2000), and the y-axis shows the proportion of spectra for which the set of de novo predictions contained a correct sequence.

In Figure 6.8 we also see that reranking de novo sequences significantly increases the accuracy rates. There is an increase of 15-20% when considering small sets of predictions (1-10 sequences). The performance gap is still very significant for 50 and 100 predicted sequences, were the ranked PepNovo results practically reach the maximum they can attain (which is the value for the regular PepNovo results at 2000 sequences). With sets larger than 100 sequences, the gap naturally narrows until the two curves meet at 2000. These results show that the reranking is capable of taking correct, but poorly scoring sequences, and move them ahead in the ranks. Often the sequences

Table 6.1: Average prediction lengths in de novo benchmarking experiments. For each dataset we note the average length of the top-ranked correct predictions. [a] The average length of the peptides in the OPD280 dataset was 10.5 amino acids, and in the ISB769 dataset it was 11.7 amino acids.

	Average Predicted Length	
Algorithm	OPD280 (10.5)[a]	ISB769 (11.7)[a]
PepNovo	10.2	10.7
PepNovo + Ranking	8.6	9.3
MS-Dictionary - tryptic	10.4	11.7
MS-Dictionary	10.5	11.8
Peaks	10.2	11.4

that get pushed forward are shorter than the top-scoring sequences returned by PepNovo. This phenomenon is especially common with spectra of peptides that have poor fragmentation. In such cases, the spectrum graph contains only a partial subpath that corresponds to the correct peptide, and this path frequently gets elongated with spurious edges. Therefore, PepNovo tends to output these incorrect but higher-scoring sequences ahead of the lower-scoring correct ones. Our new ranking score is capable of detecting many of these incidents and rectify the ranks accordingly. This also explains why the average top reranked sequence tends to be shorter than the average top-ranked PepNovo sequence (see Table 6.1 for the average prediction lengths of the different algorithms).

To rule out the possibility that PepNovo's superior performance, both with and without ranking, could be attributed to its prediction of shorter incomplete sequences, we benchmarked the algorithms on the task of predicting complete sequences. In these experiments we discarded any prediction that did not span the entire mass range (this applies only to PepNovo and Peaks, since MS-Dictionary always predicts complete peptides). Note, that this puts PepNovo in a slight disadvantage compared to MS-Dictionary, since PepNovo's spectrum graph is not likely to contain a complete path for poorly fragmented peptides, while MS-Dictionary's search space includes all possible peptides.

We created several test sets taken form the HEK293 dataset [218] in which all spectra were generated from peptides with specific lengths: 8,10, and 12 amino acids. Figure 6.9 depicts the results of these experiments. For each peptide length, PepNovo's

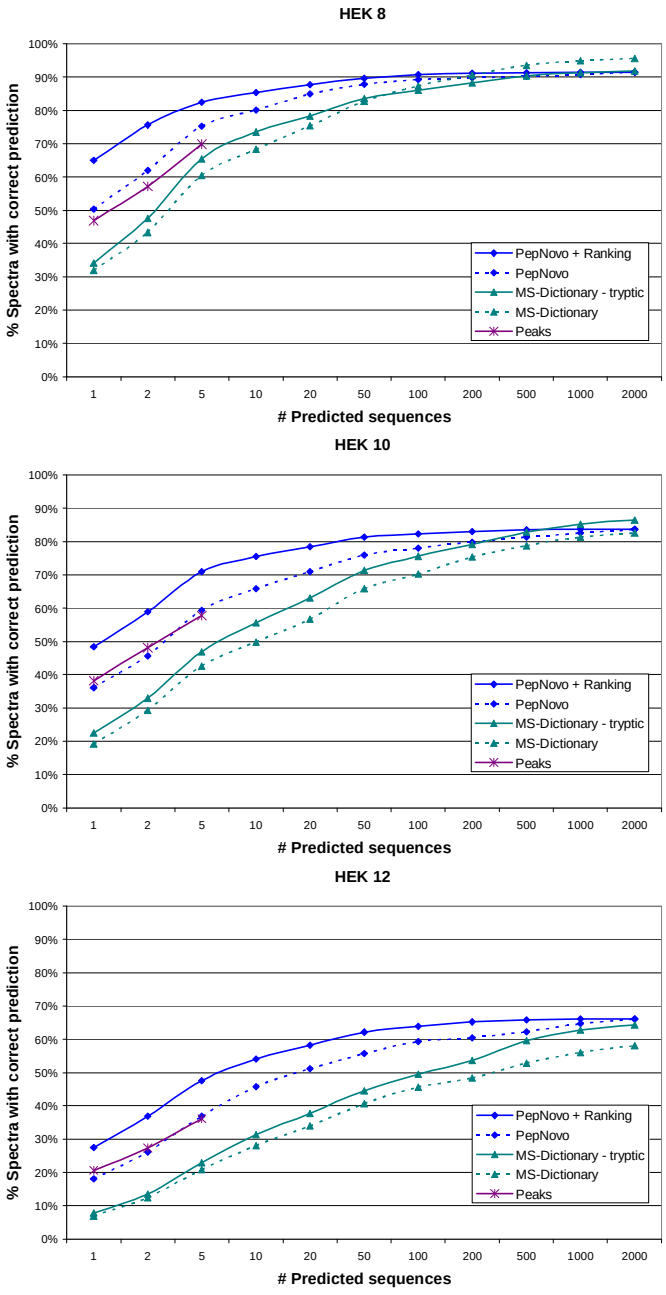


Figure 6.9: Benchmark results for sets HEK8,HEK10 and HEK12. The plots show results for PepNovo (with and without reranking), MS-Dictionary (with tryptic only and non-restricted predictions), and Peaks. In each plot the x-axis shows the size of the set of highest scoring predicted sequences (1-2000), and the y-axis shows the proportion of spectra for which the set of de novo predictions contained a correct sequence.

results are much more accurate than MS-Dictionary's (with as much as 30% more correct

Table 6.2: Benchmark results for tag generation. The table compares the sets of tags generated using PepNovo and ranking with tags generated without ranking (the LocalTag+ algorithm discussed in Section 3.B.3). Each algorithm generated sets of 1-500 tags of length 3-6 amino acids. The test set consisted of 685 spectra from the ISB dataset (see Section 3.C for more details).

Algorithm (tag length)	Number of tags								
	1	3	5	10	25	50	100	250	500
LocalTag+ (3)	0.752	0.828	0.853	0.893	0.927	0.945	0.959	0.965	0.974
PepNovo + ranking (3)	0.772	0.886	0.909	0.933	0.949	0.962	0.968	0.985	0.985
LocalTag+ (4)	0.676	0.772	0.804	0.844	0.891	0.914	0.930	0.950	0.959
PepNovo +ranking (4)	0.728	0.850	0.872	0.892	0.915	0.940	0.949	0.956	0.964
LocalTag+ (5)	0.578	0.670	0.707	0.782	0.844	0.866	0.879	0.915	0.930
PepNovo+ranking (5)	0.663	0.793	0.828	0.850	0.880	0.893	0.908	0.927	0.940
LocalTag+ (6)	0.502	0.603	0.657	0.724	0.784	0.806	0.828	0.850	0.872
PepNovo+ranking (6)	0.587	0.720	0.750	0.803	0.840	0.872	0.880	0.893	0.902

sequences for small sets of predicted sequences). Only when very large prediction sets are examined (200 sequences with length 8 and 500 sequences for length 10), does MS-Dictionary catch up with PepNovo's performance. These additional identifications made by MS-Dictionary belong to poorly fragmented peptides that do not have complete paths in PepNovo's spectrum graph. It is likely that PepNovo would be able to predict correct partial sequences in these cases.

PepNovo's performance without ranking is at par with Peaks (Peaks has slightly better performance for length 8, while PepNovo has better performance with lengths 10 and 12). However, when PepNovo's results are reranked using our new scoring function, PepNovo exhibits a significant performance boost. The accuracy of the top predicted sequence rises by 10%-15%, and this gap is maintained even when we examine sets of 5, which is the maximal number of sequences generated by Peaks for each query spectrum.

6.C.3 Benchmark Results For Tag Generation

Our scoring function is not restricted to ranking long de novo sequences. It can also be used to rerank lists of tags, and as we show below, can be quite useful in creating covering sets of tags. Since the characteristics of short sequence tags are much different than longer de novo sequences, we created special ranking model for each specific tag

length from 3 to 6 amino acids. Generating tags was done in similar fashion to the LocalTag+ method described in Section 3.B.3. To generate x tags of a given length, we we used PepNovo and extracted $4x - 6x$ tags. A small number of these tags came from parsing the highest scoring de novo paths, while the majority were directly extracted from the spectrum graph. We then used the ranking score to rerank the lists and return a set of x tags.

Table 6.2 holds results of benchmark experiments in which we compared the performance of our new tagging with the LocalTag+ algorithm. The ranking procedure shows a clear superiority for all lengths examined, though for the shorter tag lengths the advantage diminishes somewhat when we look at large sets of tags. The table also shows that if one is concerned about the tagging efficiency, using larger tags can be quite advantageous in reducing the number of database hits. For instance, a tag of length 6 is about 400 times more efficient for filtration than a tag of length 4. However, using 100 tags of length 6 gives an 88% chance that the predicted set of tags contains a correct sequence, while using a single tag of length 4 only has a 73% chance. The problem with relying solely on long tags is that many peptides have poor fragmentation patterns. In these cases, the spectrum graph often does not contain a correct tag of the desired length, or the correct path has such a poor score, that it does not get into the initial set of tags. This is evident in the table where the results for tags of length 3 and 4 reach 96%-98% while the tags of length 6 reach only 90%.

We found that in order to make tags both efficient and accurate we should use a mixture of tags. For instance, by default, InsPecT generates 25 tags of length 3 for each query spectrum. Since increasing the tag length by one amino acid gives about 20 times higher filtration efficiency, using 100 tags of length 5 would make the database filtration $\approx 100X$ more efficient. According to the table, this gives correct results in 88% of the test cases. However, we can get superior results if we select tags with several lengths. for instance, if we use a mixture of tags 3 of length 4, 35 of length 5, and 100 of length 6, this too has the same $100X$ increase in efficiency, but generate a correct set of tags in 93.1% of the cases. This accuracy rate is even higher than the 92.7% that is obtained when using the one hundred times less efficient set of 25 tags of length 3 that is generated by LocalTag+. Note that when we select this mixed set of tags, we need to

eliminate redundancies that arise when we use a long tag that is "covered" by a shorter one; otherwise, the tagging becomes less accurate.

LocalTag+ has an advantage over the new ranking-based method when it comes to running time. LocalTag+ needs 0.05-0.1 seconds to generate tags, while tags obtained by ranking take about 10 times longer. This means that for simple searches (e.g., small databases or no PTMs), using LocalTag+ will probably give the fastest overall running time. However, when the time required for tag generation is dwarfed by the database scanning time, such as when performing blind searches or searches involving large genomes, it can be quite beneficial to use the longer tags generated by PepNovo with ranking.

6.C.4 Scoring Database Search Results

In this section we present experimental results that demonstrate that our ranking score improves the performance of a database searches. Since the requirements of scoring database search results are different than scoring de novo sequences (see Section 6.A.1), we trained new models specifically optimized for scoring database search results. This training was performed slightly differently than the method we used for de novo ranking (described in Section 6.C.1). Instead of using incorrect de novo predictions for false peptide-spectrum matches, we used incorrect database search results (obtained from a run against a large set of shuffled protein sequences). This was done because the search space in a database search is much smaller than the space of all peptides, and thus generates "weaker" incorrect peptide predictions. In addition, we selected the training pairs of peptide-spectrum matches a bit differently. Instead of having 100% of the pairs of instances of spectrum-peptide matches come from the same spectrum (as was the case for scoring de novo sequencing), we found that optimal results were obtained when only 20% of the pairs were selected this way. The remaining 80% were pairs of matches from different spectra (i.e., we added instances to the model's feedback function that ranked a correct peptide-spectrum match of spectrum S ahead of an incorrect peptide-spectrum match of spectrum S'). This was done to give a higher weight the goal of an ideal database scoring function, which brings correct matches ahead of the incorrect matches from *all* other spectra, as opposed to the goal of a de novo scoring

function that is just required to bring the correct match ahead of the incorrect matches from the *same* spectrum (see Section 6.A.1).

To benchmark the performance of our new scoring method we chose a run from the HEK293 dataset [218], consisting of 750000 spectra. We used InsPecT [219] to perform the database search against the IPI human protein sequence database (version 3.42 containing $\approx 30M$ amino acids). The searches were conducted in three different modes:

- **Regular InsPecT** - The default mode for running InsPecT (relies on InsPecT's tagging and scoring functions).
- **InsPecT Tags + Rank Score** - We take the regular output from InsPecT which supplies 10 candidate peptides per spectrum and rescore them using our ranking score function.
- **PepNovo Tags + Rank Score** - We supply InsPecT with a set of tags generated by PepNovo. Inspect then finds for each spectrum the 10 top scoring peptides (using InsPecT's scoring function). We then post-process these results and rescore the peptides using the ranking scoring function.

The final post-processing step of the database search (in all three modes), was to filter the results to maintain a false discovery rate of 1% at the spectrum level which corresponds to approximately 4% at the peptide level (i.e., 1% of the reported spectrum identifications and 4% of the reported peptide identifications are expected to be false positives).

The tags generated by PepNovo (used only in the "PepNovo Tags + Rank Score" search) were a mixture of 3 tags of length 4, 35 tags of length 5 and 100 tags of length 6 (as described above in Section 6.C.3). This mixture of tags is 100 times more efficient than the tags used by InsPecT's regular search (25 tags of length 3). Since there are many fixed-time operations involved with the database search (file I/O, scanning the DB, etc.), there is not a direct linear relationship between the tagging efficiency and the actual run-time speedup. Thus, PepNovo's 100 times more efficient tags led to an approximately 15-fold reduction in InsPecT's run-time.

Table 6.3 reports the results of these benchmark experiments. The table shows the number of spectra and peptides identified with each of the three search modes, and

Table 6.3: Database search results for a HEK293 run (750000 spectra) against the IPI sequence database (version 3.42). The table compares the results obtained by using Inspect in the default mode ("Regular Inspect"), rescoring Inspect results ("Inspect Tags + Rank Score"), and using PepNovo tags and rescoring results ("PepNovo Tags + Rank Score"). The identifications were made with a false discovery rate of 1% at the spectrum level which is approximately 4% at the peptide level. The values in parentheses indicate the relative gain in identifications compared to the regular InsPecT search.

Identifications	Search type				
	Regular InsPecT	InsPecT Tags + Rank Score		PepNovo Tags + Rank Score	
Spectra:					
Charge 1	6891	10017	(+45.3%)	13134	(+90.5%)
Charge 2	89259	96244	(+7.8%)	99775	(+11.8%)
Charge 3	14284	19516	(+36.6%)	18324	(+28.3%)
Total	110434	125577	(+13.7%)	131233	(+18.8%)
Peptides:					
Charge 1	3961	5721	(+44.4%)	6977	(+76.1%)
Charge 2	20304	22061	(+8.7%)	23526	(+15.9%)
Charge 3	3217	4586	(+42.5%)	4450	(+38.3%)
Total (unique)	22518	25827	(+14.7%)	27685	(+22.9%)

breaks the results down according to charges. The total number of identified peptides is lower than the sum of the identifications made with charges 1-3 because often the same peptides were identified by several spectra with different charges, and we reported only the number of unique peptide identifications. The maximal number of identifications was obtained using the method "PepNovo Tags + Rank Score" (18.8% more spectra and 22.9% more peptides than the regular InsPecT run). The largest increase in identifications was seen with charge 1, where the number of identified spectra rose by 90.5% and the number of peptides rose by 76.1% higher, compared to the number of identifications obtained with InsPecT. This indicates that InsPecT's scoring models do a poorer job with singly-charged peptides, compared to their handling of doubly charged ones. The results for "InsPecT tags + Rank Score" also show a considerable improvement compared to the default InsPecT run, with an increase of 13.7% in the number of identified spectra and 14.7% in the number of identified peptides. When we compare these numbers to the improvement of +18.8% spectra and +22.9% peptides obtained with "PepNovo Tags + Rank Score", we can say that almost 2/3 of the improvement of "PepNovo Tags

+ Rank Score" can be attributed to our improved scoring, while the rest of the gained identifications come from PepNovo's more accurate tags. Note that PepNovo's tags yield more identifications than InsPecT's tags despite the fact that they are 100 times more efficient. Interestingly, for triply-charged peptides, the results with "InsPecT Tags + Rank Score" are better than the results obtained with PepNovo's tags. This means that for triply-charged spectra InsPecT's tags are more accurate than PepNovo's. The reason this happens is because triply-charged peptides typically have poor fragmentation, so in many cases it is quite difficult to extract long tags (4,5 or 6 amino acids long), while still relatively easy to get a good tag 3 amino acid long.

6.C.5 Searching MS/MS Spectra Against Six-Frame Translations

Despite advances in genome sequencing and gene annotation algorithms, many genes remain unidentified even in the well-studied organisms [199, 203]. Annotation of genes using evidence of protein expression obtained via MS/MS experiments ("proteogenomic mapping") is suggested as a complementary method to sequence-based gene prediction algorithms [5, 27, 32, 52, 66, 87, 105, 113, 160, 193, 218]. Since proteogenomic studies involve searching mass spectra against all possible reading frames in a genome (a "six-frame translation"), the process can be quite time consuming when large eukaryotic genomes are investigated. In addition, the large search space encountered in proteogenomic studies leads also to lower sensitivity (fewer identifications) compared to searches against smaller protein databases [32, 36].

There have been several recent proteogenomic studies involving the six-frame translations of the human genome [52, 66, 193]. However, these studies used relatively slow search programs such as Sequest (used in [52]) and X!-Tandem (used in [66]), or relied on high-resolution FTMS to reduce the number of candidates that need to be considered [193]. In our experiments, searching a single spectrum against a six-frame translation of the human genome required approximately 5 minutes of CPU time using InsPecT, which was benchmarked as being 10 times faster than X!-Tandem and 60 times faster than Sequest [160]. In addition to being quicker, InsPecT typically makes more identifications than the aforementioned search engines. Recently, Kim et al.[120] developed the MS-Dictionary algorithm which is capable of performing rapid searches

(on the order of a second per spectrum) against large sequence databases. However, the feasibility of this approach was demonstrated only on a small subset of the MS/MS datasets (they focused their identifications on doubly-charged peptides of length 10-14).

In this section we demonstrate how our novel ranking score helps alleviate the two main deficiencies of proteogenomic mapping: speed and accuracy. Using PepNovo-generated tags we are able to perform the database search significantly faster than the current state-of-the-art ($\approx$ 15 times faster than InsPecT). In addition, reranking the results using our new scoring function significantly increases the number of identified peptides compared to the results obtained by a regular run of InsPecT.

We performed a benchmark experiment using the same HEK293 run used above to search the IPI sequence database. This time our sequence database was a six-frame translation of the human genome (NCBI build 35.3 masked using RepeatMasker), which contained approximately 3 billion amino acids – one hundred times larger than the IPI sequence database. The sequences in the six-frame translation were split into 40 files to facilitate the InsPecT runs. Each sequence file was searched separately, and the results were then pooled and filtered, keeping the ten highest scoring peptide identifications for each spectrum. Similarly to the experiments with the IPI database, we ran three different types of searches: "Regular InsPecT", "InsPecT Tags + Rank Score", and "PepNovo Tags + Rank Score".

Table 6.4 reports the results of these benchmark searches against a six-frame translation of the human genome. Similar to experiments with the IPI database, we see a significant improvement when using PepNovo's tags and the new ranking score. However, with the six-frame translation's challenging one hundred-fold increase to search space size, the advantages of our new scoring become much more significant. There is a 61.3% increase in the number of identified peptides with "PepNovo Tags + Rank Score" search compared to a regular InsPecT run, and a 38.9% increase when only the reranking of results is applied. This increase is almost three times larger than the increase observed when we searched the IPI sequence database. Note, that the total number of peptides identified in the six-frame translation is significantly lower than the number identified when searching IPI. However, while the regular InsPecT search losses 55% of its peptide identifications (it goes from 22518 peptides identifications down to 10356),

Table 6.4: Database search results for a HEK293 run (750000 spectra) against a six-frame translation of the human genome (NCBI build 35.3 masked using RepeatMasker). The table compares the results obtained by using Inspect in the default mode ("Regular Inspect"), rescoring Inspect results ("Inspect Tags + Rank Score"), and using PepNovo tags and rescoring results ("PepNovo Tags + Rank Score"). The identifications were made with a false discovery rate of 1% at the spectrum level which is approximately 4% at the peptide level. The values in parentheses indicate the relative gain in identifications compared to the regular InsPecT search.

Identifications	Search type				
	Regular InsPecT	InsPecT Tags + Rank Score		PepNovo Tags + Rank Score	
Spectra:					
Charge 1	3109	5836	(+87.7%)	7268	(+133.7%)
Charge 2	39997	53107	(+32.7%)	61855	(+54.6%)
Charge 3	4529	9557	(+111.0%)	10426	(+130.2%)
Total	47635	68500	(+43.8%)	79549	(+66.9%)
Peptides:					
Charge 1	1761	3279	(+86.2%)	3820	(+116.9%)
Charge 2	9430	12326	(+30.7%)	13725	(+45.5%)
Charge 3	1020	2244	(+120.0%)	2945	(+188.7%)
Total (unique)	10356	14391	(+38.9%)	16706	(+61.3%)

the "PepNovo Tags + Rank Score" method fairs significantly better, losing only 40% of its identifications (from 27685 peptides down to 16706). These reductions in the number of peptide identifications are in-line with previous proteogenomic experiments [36]. One reason behind these reductions in identifications is because of the limited discriminatory power of the scoring functions. With a six-frame translation we encounter many more high-scoring incorrect peptide-spectrum matches compared to the number encountered when searching a significantly smaller search space. This reduces the number of positive identifications that can be accepted at a given false positive rate. Castellana et al.[27] witnessed a 30% reduction in the number of identified peptides that fell within exonic regions when switching from a protein sequence database to a six-frame translation of the genome of *Arabidopsis thaliana*. In addition, many of the peptides identified when searching protein sequence databases happen to fall on exonic boundaries. These products of splice events are not present in six-frame translations, and are bound to be missed. The number of such cases can be surprisingly large. For example, Kim et al.[120]

Table 6.5: Comparison between identifications made with IPI and six-frame searches. The table compares the results obtained by using InsPecT in the default mode ("Regular Inspect"), rescoring Inspect results ("InsPecT Tags + Rank Score"), and using PepNovo tags along with rescoring of the results ("PepNovo Tags + Rank Score").

Search Type	# Peptides identified when searching against IPI	# Peptides identified when searching six-frame translation	# Peptides identified in IPI that were not in the six-frame translation DB	# Peptides from IPI that were in six-frame DB, but lost due to deficient scoring	# Novel peptides identified only in six-frame search
Regular InsPecT	22518	10356	7684	5103	625
InsPecT Tags + Rank Score	25827	14391	9227	3037	828
PepNovo Tags + Rank Score	27685	16706	9683	2449	1153

found that 36.4% of the identifications made when searching MS/MS spectra against the human IPI sequence database belonged to peptides that spanned exonic boundaries.

Table 6.5 compares between the peptide identifications made searching against IPI with the identifications made searching against the six-frame translation of the human genome. The fourth column shows that with all three search methods, a little more than a third of the peptides identified in the IPI search are lost because they do not appear in the six-frame translation (since they span exonic boundaries).

Many peptides that were identified in IPI and were also present in the six-frame translation, were not included in the final set of positive identifications form the six-frame search. The culprit in these cases was the larger search space, which greatly increased the number of high-scoring incorrect peptide-spectrum matches. These additional high-scoring incorrect matches raised the scoring bar needed to be accepted as positive identifications (at the same false discovery rate). However, ultimately these loses can be attributed to deficiencies in the scoring functions since the more powerful and discriminating the scoring function is, the less we expect to experience this type of identification loses. From this perspective our novel ranking score performs significantly better than the default InsPecT scoring function. The fifth column shows that while a regular InsPecT run lost 5103 of the 22518 peptides it identified in IPI (22.7%), rescoring the InsPecT results using our novel score reduced this loss to 3037 from 25827 (11.8%).

Interestingly, the search that used PepNovo tags lost only 2449 of the 27685 identified peptides (8.8%). The reason PepNovo's tags lose fewer peptide identifications is that these tags are 100 times more efficient than the tags used by InsPecT. This higher filtration rate results in fewer high-scoring incorrect matches, which ultimately lowers the score threshold required to accept positive matches.

The last column in Table 6.5 lists the number of peptide identifications that are unique to the six-frame translation, representing products of unannotated genes. All three search methods show a similar ratio of these peptide identifications (between 5.7% and 6.9%). However since many more peptides got identified with "PepNovo Tags + Rank Score", the number of novel peptides found with this method (1153), is significantly higher than the number found using a regular InsPecT search (625) or a rescored InsPecT run (828).

6.D Discussion

In this chapter we explored how discriminative data-driven ranking models could be used successfully for the complex task of scoring peptide-spectrum matches. In the past, generative machine learning methods were typically used for this task. We argued that this scoring problem is inherently a ranking problem – we need to bring the correct peptide-spectrum match ahead of the incorrect ones; it is less natural to treat this as a classification problem. We had at our disposal a large set of diverse features which described many aspects that are known to be indicative of strong or poor peptide-spectrum matches. These features come from a diverse set of sources: the peptide's path in the spectrum graph, peak annotations (e.g., the numbers of b, y-ions that got annotated), the peptide's sequence (characteristics of proteotypic peptides), and others. An important source of features were the peak rank predictions we created using the models developed in Chapter 5. Each of these features by itself might be only marginally successful at discriminating between a good and bad peptide-spectrum match, and thus constitutes what is known as a "weak learner". The RankBoost algorithm [77] proved to be a very effective tool for combining this diverse set of features into powerful discriminating scoring functions. In addition, from the models created by RankBoost

we were able to gain insight into the dynamics and contributions of the various features, unlike models from other popular learning algorithms which are basically "black boxes" (e.g., support vector machines or neural networks).

We designed our models to be able to offer a general stand-alone scoring function that requires as input only a peptide sequence and the experimental mass spectrum. This makes our scoring function applicable to data obtained from a diverse set of experimental platforms and protocols. However, given additional platform and experiment-specific information, our scoring function could be made even more discriminating. For example, using immobilized pH gradient isoelectric focusing as a first-dimension separation can increase the discriminatory power of the scoring function by incorporating features that measure the difference between a peptide's observed and predicted isoelectric point. This feature can cause the removal of a large portion of the spurious peptide-spectrum matches [26, 123]. Similarly, comparing between the observed and predicted peptide retention times can also help remove many candidate peptides from consideration [7, 124, 166, 240]. We could make better use of the proteotypic peptide properties by using detectability scores that require additional information such as the spacial location of the peptide in the protein's 3D structure [215].

We demonstrated how our novel scoring function can be used to deliver superior performance in several MS/MS scoring tasks. By reranking the original order of the de novo results according to our novel rank score we were able significantly improved PepNovo's accuracy rates. This boosted the algorithm's performance well above the current state-of-the-art. For instance, when making a single sequence prediction, our reranked results are 10%-20% higher than the current high-performance algorithms (PepNovo and the commercial software Peaks). This performance gap persists even for larger sets of predictions (see Figures 6.8 and 6.9).

Our novel score also greatly enhanced the accuracy of PepNovo's peptide sequence tags (see Table 6.2). This enabled us to generate longer tags without compromising their ability to be a covering set (i.e., a set which contains at least one completely correct tag). The enhanced tagging capability both increased the accuracy of database searches and significantly reduced the running time, enabling us to speedup InsPecT's searches against a six-frame translation of the human genome by a factor of 15.

In addition to developing scoring models for de Novo sequencing, we trained specific models for scoring database search results. When used to rescore InsPecT runs that searched MS/MS spectra against the human IPI protein sequences ($\approx$ 30 million amino acids), our new score was able to increase the number of peptide identifications by 14.7%. The increase grew to 22.9% when the search used PepNovo's tags instead of the ones generated by InsPecT. However, the benefits of our novel scoring method were more substantial when applied to results of a search against a six-frame translation of the human genome ($\approx$ 3 billion amino acids). Using our models to rescore InsPecT's results led to a 38.9% increase in the number of identified peptides; using PepNovo tags along with the scoring led to a substantial increase of 61.3%. With our novel score we also almost doubled the number of novel peptide identifications belonging to unannotated genes (increasing the 625 new peptides found in a regular InsPecT search to 1153).

These results underscore the fact that our models perform particularly well in challenging domains that have large search spaces. This trait becomes especially important when we start to consider more and more complex analysis tasks, such as searches that consider alternative splicing [218], large-scale blind searches [217, 223], and even searches for fusion proteins [152]. The search spaces in these domains can be so large, and contain so many high-scoring incorrect peptide-spectrum matches, that without powerful discriminating scoring functions it will be impossible to accept but a handful of the highest-scoring, and most obvious, identifications. Many interesting identification will be lost since they will lack statistical significance with current scoring methods. Our scoring function, which can be used as a stand-alone post-processing operation, can help increase the number of interesting discoveries made in such experiments.

7

Clustering Millions of Mass Spectra

7.A Introduction

Tandem mass spectrometry (MS/MS) experiments often generate millions of spectra that can be used to identify thousands of proteins in complex samples. Analyzing such large datasets poses a computational challenge. The most common computational approach is to search spectra against a protein database [44, 61, 84, 162, 197, 219]. However, even fast algorithms which employ tag based [75, 139, 147] database filtration (used by InsPecT [219] and the Paragon algorithm [197]) or two-pass database reduction (used by X!Tandem [44]), still reach a computational bottleneck when analyzing millions of spectra against large protein databases, particularly when mutations and unexpected post-translational modifications (PTMs) are considered.

Typically in MS/MS analysis, each mass spectrum in the dataset is searched against a sequence database. At times this can be very inefficient since MS/MS datasets contain many redundancies (it is common for peptides to get selected for fragmentation more than once [211]). When mass spectra are collected from several runs, such redundancies can add up to hundreds and even thousands of spectra from the same peptide. Instead of repeating the identification process for each spectrum, it can be beneficial to perform this process once and apply the results to all similar spectra. Tabb et al. [211]

demonstrated how clustering can speed-up the analysis of single runs (though at the cost of losing some peptide identifications). This approach was later improved with the MS2Grouper algorithm [214] which was able to reduce the number of spectra that have to be searched by 20% with a reasonable trade-off of just 1% reduction in number of peptides identified when run on datasets of ≈ 50000 spectra. Beer et al. [15] developed the Pep-Miner clustering algorithm and applied it to datasets of $\approx 500,000$ spectra. They demonstrated how clustering improves analysis by reducing the runtime and generating additional peptide identifications. However, Pep-Miner (developed at IBM) is not publicly available, and little information was given on its clustering performance. Pep-Miner also relies on retention time prediction for clustering quality assurance, which can be difficult to calibrate when multiple MS runs are being clustered, unless the runs are carefully aligned [178].

Recently, researchers have tried to adapt new algorithmic ideas, first developed in the context of Internet and database clustering, to MS/MS clustering. Ramakrishnan et al. [174] and Dutta and Chen [55] proposed to use metric space embedding for MS/MS database search and clustering. While these promising approaches offer a potential solution to the problem of clustering very large datasets, the applications of these new ideas were illustrated only with a related task of filtering candidates for database searches [174] or for clustering with relatively small spectral datasets [55].

Due to the nature of MS/MS clustering, the choice of pre-processing parameters, measures of spectral similarity, and construction of cluster representatives are no less important than the speed of the clustering algorithm. For example, a fast clustering algorithm generating low-quality clustered spectra (as compared to the quality of the non-clustered spectra) is not very useful for MS/MS database searches. We developed a simple and effective MS-Clustering algorithm which is designed to rapidly process large MS/MS datasets (even in the excess of ten million spectra), while insuring the high quality of the resulting clusters. MS-Clustering reduces the number of spectra that have to be searched by up to 90% without reducing the number of identified peptides and proteins (and in many cases even increasing the number of identifications). The number of spectra identified when a clustered dataset is searched is much higher than the number of identifications made with a standard search of non-clustered data (for

large datasets the number can be doubled). This increase can be attributed to many weak spectra that do not get identified in a database search, but get identified indirectly with clustering because "spectrum vs. spectrum" analysis has some advantages over the traditional "spectrum vs. peptide" analysis. Particularly, it is difficult to predict the intensities of peaks in a theoretical spectrum (comparison with a theoretical spectrum is the basis of several MS/MS database search algorithms). Often a spectrum will show higher similarity to another experimental spectrum of the same peptide than it shows to the peptide's predicted theoretical spectrum. Thus, the spectrum can get identified via its cluster membership even though it does not get identified in a database search (this principle of similarity between experimental spectra is the basis for the spectral library approaches to peptide identifications [46, 81, 125, 131, 204, 239]). For this reason clustering also reduces the number of false database identifications with low-quality spectra (a low signal-to-noise ratio is a leading cause of erroneous database identifications). By joining together both high-quality and low-quality spectra of the same peptide, we decrease the probability of making erroneous identifications as a result of searching the low-quality spectra separately.

Another benefit of clustering is that it can help focus a researcher's efforts when selecting candidates for advanced time-consuming searchers. For example, while it is possible to identify spectra of peptides with mutations, single amino acid polymorphisms, and unexpected PTMs using "blind" PTM searches [223], such searches against large databases become rather time-consuming. By restricting this advanced search to the set of unidentified clusters, we can reduce the computational time required for advanced analysis. Finding large unidentified clusters can also point us to interesting cases that are not identified in existing database searches such as programmed frameshifts or DNA sequencing errors [87].

7.B Materials and Methods

7.B.1 MS/MS Datasets

We used three MS/MS datasets generated from samples of different organisms to analyze our algorithm's performance (see references for complete details on the pro-

tocols used to generate the data).

- **Human** [218] - 11.4 million spectra from 14 runs from samples of the HEK293 cell culture. Spectra were acquired on an LTQ linear ion trap tandem mass spectrometer. The sequence database used to identify proteins was human IPI (version 3.18, 26.7M amino acids). In addition to performing experiments on all 14 runs, we selected a single run (793000 spectra) and a subset of five runs (4 million spectra) for our experiments, in order to evaluate how increasing the number of runs affects the clustering and identification performance.

- **Shewanella** [87, 143] - 14.5 million spectra from multiple samples of *Shewanella oneidensis* MR-1. The majority of the spectra were generated on ion-trap mass spectrometers, while approximately 2 million mass spectra generated by an FT-ICR mass spectrometer. The sequence database used to identify proteins was downloaded from NCBI (release 20070113, 1.45M amino acids).

- **Dictyostelium** [218] - 1.4 million spectra from samples of light-chain, heavy-chain, and undefined cells of *Dictyostelium discoideum*, acquired on an LTQ linear ion trap tandem mass spectrometer. The sequence database used to identify proteins was downloaded from Dictybase.org (release 20060828, 7.36M amino acids).

- **Yeast** [22] - 179377 spectra from samples of *Saccharomyces cerevisiae*, acquired on an LCQ-Dexa XP ion-trap mass spectrometer. We used 3 small runs with different experimental settings: nanoLC-LC MS/MS (MudPIT), nanoLC-MS/MS with gas phase fractionation by mass range selection, and nanoLC-MS/MS with gas phase fractionation by ion abundance selection. The sequence database used to identify proteins was downloaded from SGD (release 20070112, 4.94M amino acids).

7.B.2 Database Search

We used the InsPecT database search tool [219] to perform peptide identifications (release 20070613), using the default search parameters (precursor mass tolerance 2.5 Da, fragment ion tolerance 0.5 Da). All searches were performed using a shuffled decoy database. When computing Inspect F-scores, the files from each experiment were

pooled together (rather than analyzing them in a run-by-run fashion). The InsPecT F-score threshold values for accepting identifications were selected to ensure a true positive peptide identification rate of 98% (i.e., only 2% of the peptide hits came from the decoy database).

7.B.3 Filtering MS/MS Datasets

Large MS/MS datasets contain many low-quality spectra that cannot result in reliable peptide identifications [19, 150]. Typically, when a whole MS/MS dataset is searched, only a small fraction of the spectra (less than 20%) get identified. Many low-quality spectra have characteristics that distinguish them from identifiable spectra (lack of complimentary b/y peak pairs, lack of peptide sequence tags, etc.) which can be used by classification algorithms to identify these spectra [19, 70, 150, 172]. Removing such spectra is beneficial to clustering performance since it reduces the number of spectra that undergo pairwise comparisons. Furthermore, filtering reduces the number of clusters generated by the algorithm that get submitted for further analysis. We performed spectral quality filtering as a pre-processing step using our in-house software MS-Filter (available from `http://peptide.ucsd.edu`). MS-Filter uses an approach similar to the one described in ref. [150] and complements it by charge selection, and precursor mass correction. The filtering procedure typically requires $\approx$ 5 milliseconds per spectrum. We ran all experiments with the default quality threshold values. Though filtering can lead to the exclusion of some identifiable spectra (less than 0.5%, as benchmarked at the default values), filtering can actually increase the identification rates for a given true positive rate. For example, when searching a single run from the Human samples, filtering increased the number of spectra, peptides, and proteins identified by approximately 0.7% (see Table 7.2). The additional identifications can be attributed to the fact that when many low quality spectra are removed by the filtering, the number of spurious hits to the decoy database is greatly reduced. Thus for a given true positive rate, the score threshold required to accept an identification is lower with a filtered dataset than it is with an unfiltered one.

7.B.4 MS-Clustering Algorithm

Our MS-Clustering algorithm is similar in several aspects to the Pep-Miner algorithm [15] but has a number of optimization steps that enable analysis of over 10 million mass spectra (an order of magnitude increase in the maximum number of analyzed spectra compared to the results reported for Pep-Miner). The three major components of our approach are a spectral similarity measure, a method for the selection of a cluster's representative spectrum, and a clustering algorithm itself.

Spectral Similarity

In order to cluster mass spectra we need to determine the similarity between them. We use the normalized dot-product, which has previously been found to work well by several groups that have approached similar problems [15, 125, 131, 174, 204, 211, 214, 228].

To calculate the normalized dot-product of two mass spectra S and S', we first reduce each spectrum to a vector. Since the computation of the spectral similarity is a major part of the clustering algorithm, restricting the size of theses vectors can reduce the running time. To construct such vectors we first select the k strongest peaks from S and S' (we assume that S and S' have similar precursor masses). Joining these two sets of masses yields a set of masses $M = \{m_1, \ldots, m_t\}$, where $k \leq t \leq 2k$. M may contain less than $2k$ masses because duplicate masses are removed (we consider two peaks to have a similar mass if they are within 0.5 Da from each other). Finally, we reduce the spectrum S to a vector $s = s_1, \ldots, s_t$ by assigning to each s_i the intensity found at mass m_i in S if m_i was one of the top k peaks in S, otherwise 0 is given to that position. Similarly, we fill s' using the intensities of the peaks in S'. In our experiments we found that for these similarity computations it is optimal to set k to a value that corresponds to 15 peaks per 1000 Da of peptide mass. Once spectra S and S' are converted to vectors, their normalized dot-product is given by

$$Similarity(S, S') = \frac{\sum_{i=1}^{t} s_i \cdot s'_i}{\sqrt{\sum_{i=1}^{t} {s_i}^2 \cdot \sum_{i=1}^{t} {s'_i}^2}} \tag{7.1}$$

The normalized dot-product takes values between 0 (when spectra do not share any selected peaks) and 1.

Dot-products were initially used for measuring similarity between mass spectra of chemical compounds, whose mass spectra typically contain a small number of peaks [204]. Directly applying this measure to spectra of peptides can yield suboptimal results since a small number of strong peaks in the spectrum can dominate the outcome of the spectral similarity computation. Scaling peak intensities has been shown to improve the quality of the similarity computations [204]. One method that has been suggested is to scale a peak's intensity according to the square root of the intensity [81, 131, 204]. The scaling method we found most suitable for our data was to first normalize the peak intensities to bring the total spectrum's intensity to 1000 and then fill the dot-product vectors with the natural logarithm of the selected peaks' intensities.

Cluster Representatives

Our algorithm generates a single spectrum representative for each cluster with more than one spectrum (singleton clusters use the spectrum itself as the cluster representative). Having a single representative is beneficial in two ways. First, it reduces the number of spectral similarity computations performed by the clustering algorithm (computing spectral similarity of a candidate spectrum to a cluster requires only a comparison with the cluster's representative and not the individual cluster members). Second, a single cluster representative can be submitted for the analysis and the results can be assigned to all cluster members.

Since "all spectra are not created equal", it helps to select representative spectra with the highest signal-to-noise ratio in the cluster or to come up with a virtual spectra with high signal-to-noise ratio. Such spectra can have a significantly higher signal-to-noise ratio than typical spectra in the clusters (see refs [12, 15] and analysis above). We examined several methods for selecting a cluster's representative and chose to use a *consensus spectrum* [46, 125, 131, 214] as the representative.

Our method for creating a consensus spectrum is as follows. Given the cluster's mass spectra, we create a single merged peak list for all the spectra, and sort the list according to the peaks' masses. The list is then scanned and when a pair of adjacent peaks having a mass difference below a specified tolerance is detected, the peaks are consolidated to a single peak with a mass that equals the weighted average of the joined

peaks' masses and an intensity that equals the sum of the joined peaks' intensities. To increase the accuracy of the peak joining, the process is repeated several times with an increasing tolerance threshold (the final threshold we used was 0.4 Da). This is done to avoid erroneous peak merging due to isotopic peaks, etc.

To increase the peptide's signal in the spectrum, we take advantage of the fact that peaks corresponding to genuine fragments are likely to appear in many of the cluster's spectra. Thus for each peak i in the consensus spectrum, we take note of the number peaks from the original spectra that were merged to create i and divide it by the total number of spectra to obtain the peak probability p_i. We then multiply the peak i's intensity by a scaling factor $\alpha = 0.95 + 0.05 * (1 + p_i)^5$. This function gives α a value close to 1 for peaks with low probability, but increases as the probability nears 1 to a maximal value of 2.55. Finally the list of peaks in the consensus spectrum is filtered using a sliding window to filter out weak peaks (in our experiments we kept the top 5 peaks in a window of 100 Da).

To determine if the consensus spectrum is the best choice, we examined the quality of five alternatives for a cluster's representative.

1. "best spectrum": the spectrum that maximizes a certain score, e.g., percent of explained intensity or percent of explained b/y ions (this is the optimal spectrum that could be selected from amongst the cluster members).

2. "consensus spectrum": a virtual spectrum constructed by consolidating all spectra in the cluster.

3. "most similar spectrum": the spectrum that has the highest average similarity to the other cluster members [81, 211].

4. "de novo spectrum": the spectrum that has the highest score when submitted to de novo sequencing.

5. "average spectrum": a spectrum chosen from the cluster at random.

Figure 7.1 shows plots in which we examine the relation between the cluster size and the quality of different types of cluster representatives. The plots were generated from 250 clusters each containing at least 100 spectra from the Human dataset which

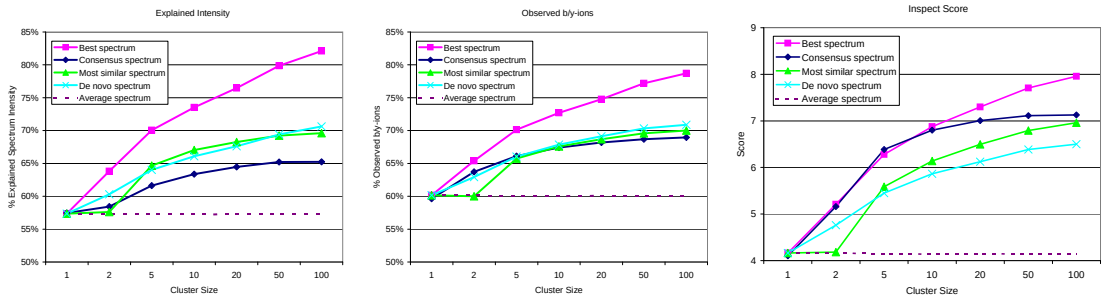


Figure 7.1: Cluster size and spectrum quality. Clusters of various sizes were evaluated to determine the fraction of explained spectrum intensity (left), proportion of observed b- and y-ions (center), and score given to the spectrum by Inspect (right). With each cluster these statistics were collected for five different cluster representatives: 1) The best spectrum, 2) The consensus spectrum, 3) The most similar spectrum, 4) The best de novo spectrum, and 5) The average spectrum.

were identified with high confidence by InsPecT. The spectra were filtered using a sliding window to maintain a peak density of approximately 50 peaks per 1000 Da of peptide mass. For each cluster size, we repeatedly drew random subsets (clusters) varying in size from 1 to 100, taken from the spectra of the original 250 large clusters. For each drawn cluster of spectra corresponding to a peptide P, we examined the percent of explained intensity (i.e., the sum of the intensities of peaks belonging to fragment ions of P), the proportion of P's b- and y-ions that were observed in the spectra and the score given to the spectrum by InsPecT when annotated with the peptide P. These three statistics were recorded for five different methods for selecting cluster representatives.

Figure 7.1 illustrates the benefits of selecting cluster representatives "wisely". Using representatives 2-4 gave spectra with a significantly higher signal-to-noise ratio than the "average" representative (5). The most similar spectrum and the top de novo spectrum have higher proportions of explained intensity (up to 5% more) than the consensus spectrum, but relatively similar proportions of observed b- and y-ions. However ultimately, when the spectra are submitted to a database search, the consensus spectra have higher InsPecT scores than the other methods (except for selecting the "best" which we only know how to identify after searching all cluster members). In fact with clusters of up to 10 spectra, the consensus spectra and the best spectra in the clusters have similar InsPecT scores (with a slight advantage for consensus spectra at size 5 which

Approximate Hierarchical Clustering

- Input:
  $1, \ldots, n$ // *data elements (spectra)*
  τ_{min} // *similarity threshold*
  r // *number rounds*
- Initialization:
  $\delta \leftarrow \frac{1-\tau_{min}}{r}$
  $Clusters \leftarrow \{\{1\}, \{2\}, \ldots, \{n\}\}$
  $\tau \leftarrow 1$
- **do** r times:
  $\tau \leftarrow \tau - \delta$
  for every cluster c in *Clusters*
  for every cluster c' preceding cluster c in *Clusters*
  if $Similarity(c, c') \geq \tau$
  append c to c'
  remove c
- Output: *Clusters*

Figure 7.2: A pseudocode description of the approximate hierarchical clustering algorithm used by MS-Clustering.

get a score of 6.4 compared to best spectrum's score of 6.3). We therefore decided to use consensus spectra as the cluster representatives for our clustering algorithm.

Clustering Algorithm

Many popular clustering algorithms such as $k-$means [62, 106] require an advance knowledge of the number of clusters that are being sought. However, the nature of MS/MS datasets precludes the use of such algorithms since it is nearly impossible to "guess" the number of clusters. Furthermore, the sheer size of MS/MS datasets makes this approach very time-consuming. A better MS/MS clustering method is to use a "bottom-up" approach like incremental hierarchical clustering [62, 106], which would start with clusters containing single spectra and build the clusters up by merging clusters with similar spectra.

Figure 7.2 describes a simple hierarchical clustering algorithm. The algorithm starts with the list *Clusters* consisting of all elements as singletons. There are r rounds

of clustering with a decreasing similarity threshold τ. In each round the algorithm tries to merge pairs of elements in *Clusters* with a similarity that exceeds the threshold τ. This is done by sequentially comparing each cluster c in *Clusters* with the clusters preceding it in the list. If there exists a cluster c' that is similar to cluster c in *Clusters* (similarity exceeds the threshold τ), the spectra in c are appended to spectra in c' and c is removed from the list of clusters (Figure 7.3). After r rounds, the final set of clusters is returned by the algorithm. Applying the algorithm to clustering of mass spectra is straightforward. The elements being clustered are the spectra themselves, and the function used to determine cluster similarity is the spectral similarity which is applied to the clusters' representative spectra (these consensus spectra are continuously updated as clusters are merged).

Our algorithm does not necessarily join clusters with maximum similarity, rather it joins the first ones it encounters that have a similarity above the threshold τ. However, since the algorithm consists of several rounds with decreasing similarity thresholds, it approximates the hierarchical clustering's gradual joining of clusters, in which the most similar clusters are merged first. By using this heuristic approach we are able to reduce the number of spectral similarity computations compared to traditional hierarchical clustering algorithms.

We employ additional heuristics that further reduce the number of similarity computations, and alleviate the computational cost associated with performing the clustering in several rounds. One heuristic we use evaluates how likely it is that two spectra belong to the same peptide, without explicitly computing the similarity between them. For example, spectra from the same peptides have similar sets of strong peaks: in our data, 98.2% of the pairs of spectra from the same peptide had at least one peak in common in their respective sets of the five strongest peaks. However, only 5.5% of the pairs of spectra from different peptides also have such a match in their top 5 peaks. Since testing for a common peak in the list of top 5 peaks can be done much quicker than a complete similarity computation, this heuristic can account for a significant reduction in running time by quickly eliminating the majority of the unnecessary similarity computations.

The second heuristic we use relies on the fact that our algorithm uses multiple rounds of cluster joining (with decreasing similarity thresholds τ). Instead of recom-

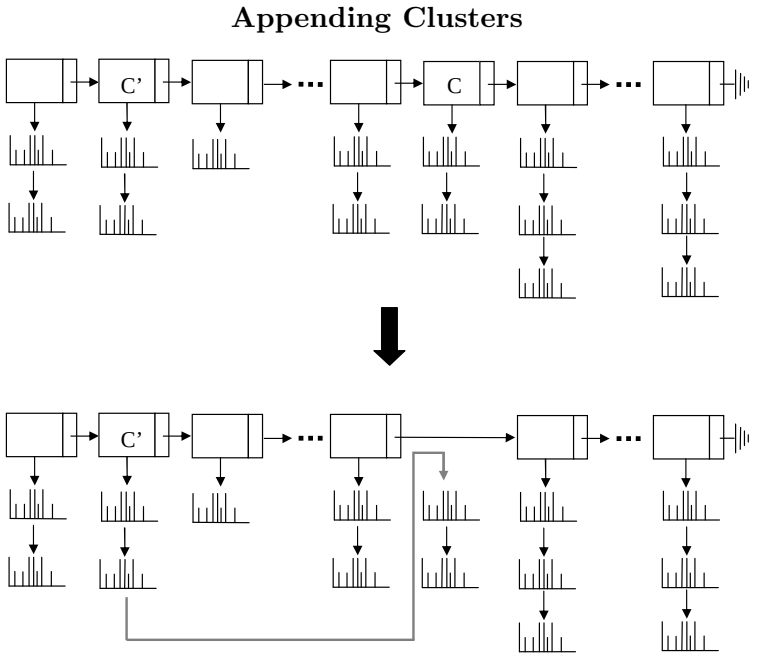


Figure 7.3: Illustration of cluster appending. The set *Clusters* is a linked list where each element is a list of spectra. When the algorithm merges cluster c with a preceding cluster c', it appends the list of spectra in cluster c to the list of spectra in cluster c' and then removes the entry for c from the linked list of clusters.

puting the similarity between pairs of consensus spectra at each round, we can carry over similarity results from one round to the next. Thus, if at a certain round a pair of clusters show extremely low similarity, we take note of this fact (by setting an appropriate indicator) and we do not examine that pair again in subsequent rounds. We use a simple bit vector to store the similarity indicators of all pairs of clusters, which for n spectra amounts to approximately $n \cdot (n-1)/2$ bits. Even when clustering large datasets (10 million spectra), the largest number of spectra simultaneously clustered is 60000, which requires 215 MB of memory to store the similarity indicators. Note that since the write operations to the bit vector always precede read operations to the same addresses, the vector does not need to be initialized at any time. This filtration heuristic can very efficient. For example, 99.9% of pairs of spectra of the same peptide have a similarity above 0.25, while less than 1% of the pairs of spectra from different peptides have a similarity that exceeds that level. Since 0.25 is a very low threshold, we can safely assume that if a pair of clusters have a similarity below 0.25 between them, even if they have additional spectra added to them in subsequent rounds, the cluster similarity will still be way below the minimum threshold for joining clusters (in our experiments the value $\tau_{min} = 0.55$ was used).

7.C Results

7.C.1 Clustering Heuristics

Table 7.1: The algorithms performance with different combinations of heuristics. The clustering algorithm was run on 0.8M spectra from the Human to evaluate the effect of adding the heuristics of carrying similarity results between the algorithm rounds and requiring pairs of spectra to have a match in their top 5 peaks. The algorithm's performance was measured both in the total number of computations performed and the total running time.

Heuristics used		# Similarity	(%)	Total	(%)
Carry Similarity	Match in Top 5	Comparisons		Run time (s)	
–	–	1.89×10^9	(100.0%)	8835	(100.0%)
+	–	4.71×10^8	(24.9%)	3731	(42.2%)
–	+	5.12×10^8	(27.1%)	4009	(45.4%)
+	+	2.26×10^8	(11.9%)	2766	(31.3%)

We first examine how the use of the clustering heuristics affects the algorithm's performance in terms of the number of similarity computations and running time. The algorithm was run with $r = 3$ rounds, a minimal similarity threshold $\tau_{min} = 0.55$, and using 15 peaks per 1000 Da for similarity computations. Table 7.1 shows that on their own, each of the heuristics approximately halved the number of similarity computations that were performed. Carrying similarity results between rounds reduced the number of these computations to 24.9% of the number of computations without heuristics, and requiring spectra to have a match in their sets of top 5 peaks reduced the number of computations to 27.1%. These two heuristics are rather complimentary to each other. The filter that requires a match of a peak in the top 5 is most effective in the algorithm's first round (in which most of the similarity computations are performed). The carrying over of similarity results between rounds is naturally only applicable to subsequent rounds. Thus, when these two heuristics are combined they produce a significant reduction in the number of similarity computations that are carried out to 11.9% of the number of computations performed when no heuristics are used. Note that calculating the similarities between all pairs of spectra in each mass bin amounts to 1.25×10^9 similarity computations. The reduction in running time is also quite impressive, using both heuristics reduces the running time less than a third of the time it takes without employing heuristics. It is worth noting that the clustering results with and without heuristics are very similar. For instance, without heuristics 71.4% of the spectra fell into non-singleton clusters compared to 70.8% when both heuristics were used.

7.C.2 Clustering Performance

The performance of the clustering algorithm depends on the similarity threshold used to determine if two spectra should be joined. A low threshold leads to large heterogeneous clusters, while a higher threshold results into a larger number of smaller, but more homogenous clusters. Table 7.2 contains the results of experiments we ran to examine the tradeoffs of using different threshold values. A single run with 793000 spectra from the Human dataset was clustered using varying similarity values between 0.35 and 1 (with a similarity threshold of 1 no clustering is effectively performed). Different thresholds should be chosen depending on the objective we wish to maximize. To

Table 7.2: Clustering performance with different similarity thresholds. Results are shown for a single run from the human dataset (793000 spectra searched against the human IPI sequence database). For each similarity threshold we report the number of spectra searched, the number of spectra identified, the number of peptides identified and the number of proteins identified. These values are compared with the values obtained from a regular non-clustered search of the same dataset (the difference is reported as a percentage).

Similarity threshold	Spectra/Clusters searched		Spectra identified		Peptides identified		Proteins identified	
Non-clustered	793000		86682		21090		6191	
0.30	167407	-78.9%	116571	+34.5%	18352	-13.0%	5772	-6.8%
0.35	204851	-74.2%	114196	+31.7%	19503	-7.5%	5991	-3.2%
0.40	241489	-69.5%	111309	+28.4%	20142	-4.5%	6096	-1.5%
0.45	276059	-65.2%	104983	+21.1%	20592	-2.4%	6178	-0.2%
0.50	309501	-61.0%	102859	+18.7%	20978	-0.5%	6229	+0.6%
0.55	340847	-57.0%	99488	+14.8%	21142	+0.2%	6282	+1.5%
0.60	369159	-53.4%	95764	+10.5%	21163	+0.3%	6275	+1.4%
0.65	394990	-50.2%	93511	+7.9%	21224	+0.6%	6266	+1.2%
0.70	417576	-47.3%	92666	+6.9%	21349	+1.2%	6300	+1.8%
0.75	436973	-44.9%	91269	+5.3%	21412	+1.5%	6310	+1.9%
0.80	452294	-43.0%	90018	+3.8%	21386	+1.4%	6289	+1.6%
0.85	467361	-41.1%	89137	+2.8%	21414	+1.5%	6286	+1.5%
0.90	478978	-39.6%	88406	+2.0%	21367	+1.3%	6268	+1.2%
0.95	487833	-38.5%	87689	+1.2%	21276	+0.9%	6245	+0.9%
1.00 (*only filtering*)	493023	-37.8%	87276	+0.7%	21239	+0.7%	6242	+0.8%

maximize the number of spectra identified, we would prefer a low threshold of 0.35-0.4 which generates large, but possibly corrupt clusters. Using threshold 0.75 maximizes the number of peptides and proteins identified (though at the expense of generating a larger number of clusters). We found that the similarity threshold of 0.55 offers both an increase in the number of identifications compared to the search of the non-clustered data (14.8% more spectra, 0.2% more peptides and and 1.5% more proteins at the same 98% rate of true positive peptide identifications), and also relatively efficient clustering (a reduction of 57% to the number of spectra submitted to database search).

Table 7.3 breaks down the identifications of spectra, peptides and proteins made in the searches described in Table 7.2. When we examine the differences between the identifications made by searching the clustered and non-clustered data we find that even

Table 7.3: Comparison of identifications in clustered and non-clustered datasets. The table contains a breakdown of the identifications of spectra, peptides and proteins as displayed in Table 7.2. For each similarity threshold the table shows how many identifications were common both when searching the clustered and non-clustered datasets, how many appeared only in the search of the non-clustered data, and how many identifications were unique to the clustered data. ([a] - Identifications common to both searches; [b] - Identifications unique to the non-clustered search; [c] - Identifications unique to the clustered search.)

Similarity	Spectra identifications			Peptide identifications			Protein identifications		
threshold	Both[a]	Non-C.[b]	Clust.[c]	Both	Non-C.	Clust.	Both	Non-C.	Clust.
0.30	64472	19194	49083	16957	4133	1395	5447	744	325
0.35	67464	16896	44410	18129	2961	1374	5672	519	319
0.40	70580	14420	39047	18854	2236	1288	5795	396	301
0.45	71294	14054	32355	19314	1776	1278	5867	324	311
0.50	73635	11956	28133	19733	1357	1245	5938	253	291
0.55	75582	10324	23130	19980	1110	1162	5987	204	295
0.60	76849	9190	18272	20186	904	977	6020	171	255
0.65	78642	7604	14433	20360	730	864	6047	144	219
0.70	80381	6030	12014	20522	568	827	6075	116	225
0.75	81721	4773	9360	20638	452	774	6102	89	208
0.80	82679	3845	7181	20698	392	688	6103	88	186
0.85	83584	3002	5457	20774	316	640	6118	73	168
0.90	84423	2203	3927	20799	291	568	6119	72	149
0.95	84887	1760	2767	20789	301	487	6118	73	127
1.00	85170	1505	2099	20787	303	452	6121	70	121

when high similarity thresholds are used, there are differences in the sets of identifications made by the two searches (though the majority of identifications are common to both). One reason why some spectra are identified only in the clustered search is that clustering greatly reduces the number of spurious hits made to the decoy database. Many weak spectra are removed from the MS/MS database search, since they cluster with stronger spectra. Consequently, this results in a smaller number of spectra that have spurious hits to the database, which leads to a lower F-score threshold for accepting identifications at a given true positive rate. For example, to maintain a 98% true positive peptide identification rate, spectra in the non-clustered data must have a minimal F-score of 3.34 to be accepted while spectra in the clustered dataset need only 3.21. There are a couple reasons why there are spectra that get identified only when searching the non-clustered dataset. First, some of the identifiable spectra are filtered out due to low quality. Second, in many cases, especially with large clusters, the consensus spectra can

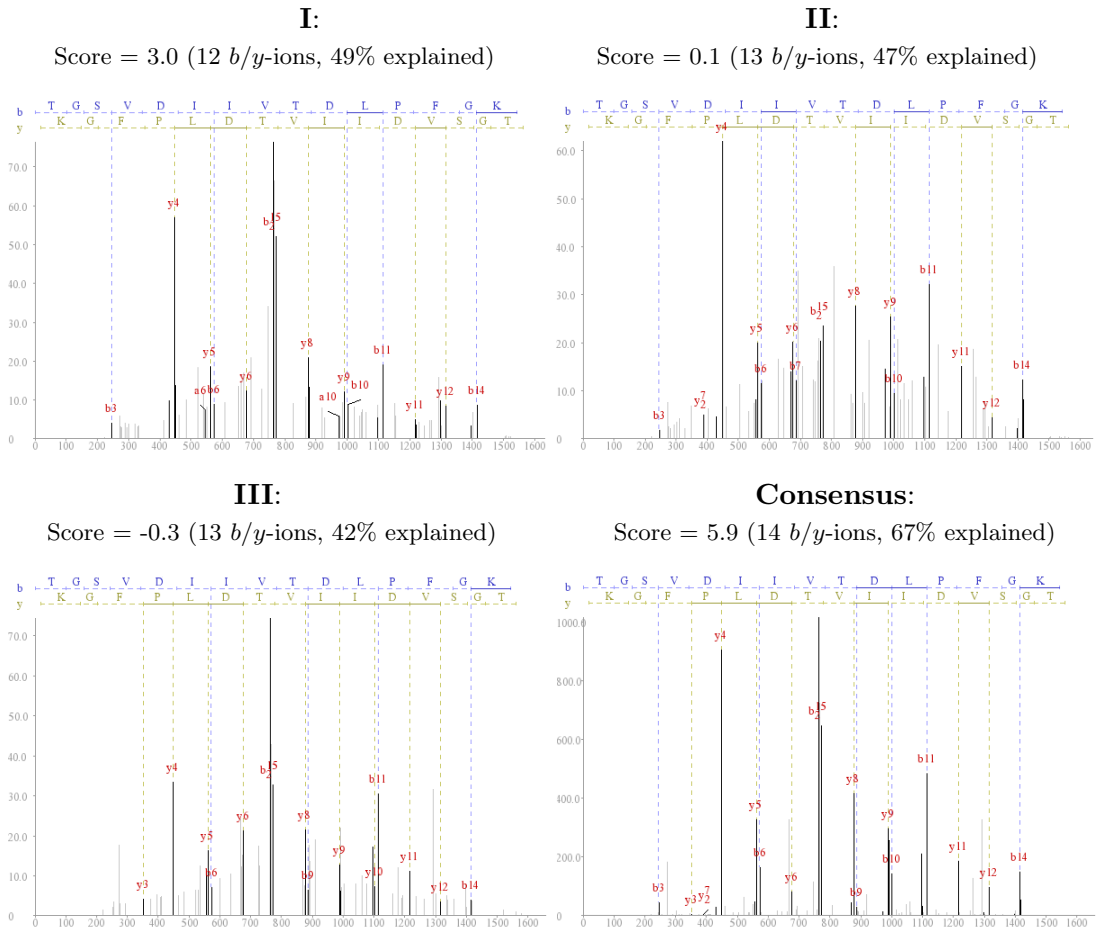


Figure 7.4: Example of cluster for the peptide TGSVDIIVTDLPFGK. A cluster of three spectra is shown along with the consensus spectrum that was created from them. For each spectrum the InsPecT score is shown, along with the number of identified b/y-ions and the percentage of the spectrum's intensity that is explained by the peptide's fragment ions. Only the consensus spectrum had a sufficiently high score to be positively identified in the database search using InsPecT. All spectra have a precursor charge 2 with precursor m/z errors below 1 Da. The figures' x-axes represents the fragments' m/z values and the y-axes represents the intensities.

have a lower signal than the best spectrum in the cluster, which can lead to it being missed (due to its lower score), while some of the individual cluster members are good enough to be identified. In any case, starting from a similarity threshold of 0.55, the total number of identifications (spectra, peptides, or proteins) made when searching a clustered dataset is higher than the number achieved without clustering.

There are some cases where clustering improves the signal-to-noise ratio beyond

the best individual members' which leads to new identifications. Figure 7.4 gives an example of a cluster of three spectra of the peptide TGSVDIIVTDLPFGK along with the consensus spectrum created from them (TGSVDIIVTDLPFGK comes from a protein sequence for which six additional peptide hits were found). Because the consensus spectrum aggregates peaks from the different spectra, it was able to accumulate peaks for 14 b/y-ions while the other spectra have peaks for at most 13 b/y-ions. However, more important is the fact that the consensus spectrum has a significantly stronger signal, explaining 67% of the spectrum's intensity, compared to between 49% (spectrum I) to 42% (spectrum III) explained intensity for the cluster members. These factors gave the consensus spectrum an InsPecT F-score of 5.9 which was sufficient to make a positive identification, while the other spectra fell short with scores between -0.3 and 3.0. When searched with Mascot [162], the three spectra had Mowse scores of 19 and below, while the consensus had a score of 31.

As the clustering similarity threshold increases, we witness a growing number of fragmented clusters i.e., several distinct clusters containing spectra of the same peptide. Though this might pose a slight increase in the computational cost since there are more spectra to analyze, cluster fragmentation is not really a problem when MS/MS data is concerned. In fact, in many cases attempting to create "optimal" clusters where all spectra of the same peptide fall into a single cluster can be counterproductive. Even with fragmented clusters, clustering still offers a significant reduction in the search time, so creating even larger clusters will only offer a modest improvement from that respect. However, an attempt to group all spectra from the same peptide into a single cluster may backfire since it may bring some noisy and unrelated spectra into the cluster yielding a noisier consensus spectrum. This can lower the number of peptides that ultimately get identified. In our experiments, a larger number of peptides and proteins were identified when we use a larger number of tighter clusters (data not shown)

In many cases fragmented clusters stem naturally from the variation observed between different experimental spectra of the same peptide [227]. Figure 7.5 shows two clusters of spectra of the same peptide VDDPNAEDKR that were not grouped together into a single cluster (three spectra are shown from each cluster). All spectra were identified confidently both by InsPecT (average InsPecT F-scores for the spectra in

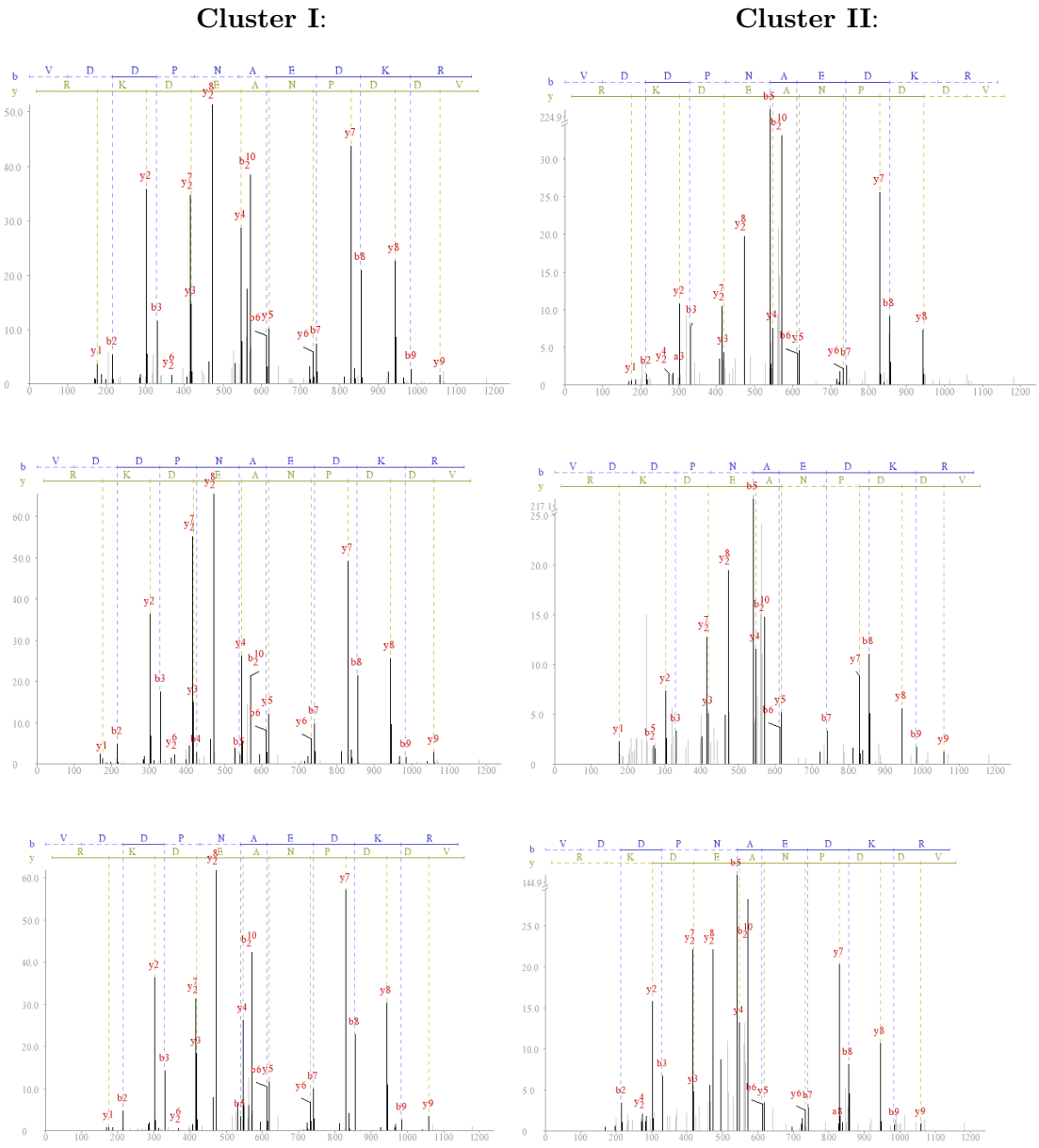


Figure 7.5: Fragmented clusters. Spectra of the peptide VDDPNAEDKR from two clusters that were not joined are shown (the figure contains 3 spectra from each cluster, originally cluster I contained 6 spectra and cluster II contained 4 spectra). The figures' x-axes represents the fragments' m/z values and the y-axes represents the intensities.

cluster I was 8.1 and 4.2 for the spectra in cluster II), and Mascot (average Mascot Mowse score of 74 for the spectra in cluster I and 52 for cluster II). There are several differences

in the fragmentation patterns between the spectra of the two clusters, the most notable difference being that in cluster I the strongest peak is the doubly charged y_8 and the b_5 is very weak, while in cluster II b_5 is the strongest peak in all three spectra. The spectra in cluster II also contain some additional noise peaks not present in cluster I. These differences were sufficient to cause MS-clustering not to join these two clusters. This example exposes a possible vulnerability of spectral libraries that use a single consensus spectrum for each peptide. If the consensus spectrum is created using examples of only one of the variants (e.g., cluster I), it is likely that spectra from the other variant (cluster II) will not be similar enough to the consensus spectrum to be identified when they are searched against the library.

7.C.3 Database Searches With Clustered MS/MS Datasets

We tested our clustering algorithm with samples of varying sizes from four different organisms. Each dataset was searched in two methods:

1. Non-clustered - regular search of complete MS/MS dataset.

2. Clustered - a search of the consensus spectra generated by MS-Clustering using the default settings ($r = 3$ rounds, a similarity threshold of $\tau_{min} = 0.55$).

Table 7.4 holds statistics on the sizes of the MS/MS datasets and the sequence databases, along with the running time required for the clustering and database searches and the total speed-up achieved by clustering. Table 7.5 holds statistics on the spectra, peptides and proteins that were identified in the experiment.

Searching a clustered dataset typically resulted in a 2×-6× speed-up in running time. Note that the database search we performed was a basic search that did not consider PTMs. With more complex searches, the speed-up achieved with clustering would be closer to the ratio in which clustering reduced the number of spectra submitted to analysis (10× in the case of the Shewanella dataset, see Table 7.5). In all cases, searching clustered datasets yielded a larger number of spectrum identifications than the non-clustered data. It ranged between a modest 9.6% gain (with the yeast samples) to almost 100% gain with the 14 runs from the human sample. This increase was achieved despite the fact that the number of cluster consensus spectra that were submitted for

Table 7.4: Running time statistics. MS/MS samples were selected from Human, Shewanella, Dictyostelium and Yeast datasets. The results are shown in two modes: without clustering, and clustering using default values. The table holds the sizes of the original MS/MS datasets and the sequence databases used for identifications. For each experiment the table also holds the run-time (in cpu hours), and the relative speed-up factor achieved by clustering vs. the non-clustered search. The runtime was measured on a 3.0GHz desktop PC with 2GB of RAM.

Dataset	DB size (amino acids)	DB size (spectra)	Experiment type	Run Time (cpu hours)			Speed-Up
				cluster	search	total	
Human (1 run)	26.7 M	0.793 M	Non-clust.	-	132.2	132.2	
			Clustered	0.77	56.6	57.4	2.3
Human (5 runs)	26.7 M	4.0 M	Non-clust.	-	664.3	664.3	
			Clustered	6.4	107.4	113.8	5.8
Human (14 runs)	26.7 M	11.4 M	Non-clust.	-	1812.7	1812.7	
			Clustered	24.3	308.2	332.5	5.5
Shewanella	1.45 M	14.5 M	Non-clust.	-	286.3	286.3	
			Clustered	26.8	28.5	55.3	5.2
Dictyo-stelium	7.36 M	1.4 M	Non-clust.	-	78.6	78.6	
			Clustered	2.3	39.3	41.6	1.9
Yeast	4.9 M	0.179 M	Non-Clust.	-	7.5	7.5	
			Clustered	0.1	3.5	3.6	2.1

analysis was six times smaller than the original dataset size. Searching clustered datasets generally increased the number of peptide and protein identifications, especially with the larger datasets, while the smaller datasets tended to display slight loss in the number of identifications.

The results in Table 7.5 illustrate an important and often neglected point that needs to be addressed when analyzing large MS/MS datasets using a decoy database. The score thresholds for spectra must be computed *using the entire set* of search results. Though it might be tempting, especially from a computational standpoint, to determine p-values or F-scores independently for portions of the dataset (e.g., analyzing each run independently when the data is collected from the instrument), this will inadvertently lead to more false positives than expected. The main culprit is that repeated runs of the same sample are highly dependent, with many of the peptide identifications appearing in multiple runs. This leads to a log-like discovery curve of new peptides (searching 14

Table 7.5: Summary of database search results. MS/MS samples were selected from Human, Shewanella, Dictyostelium and Yeast datasets. The results are shown in two modes: without clustering, and clustering using default values. The table holds for each experiment the number of spectra (or clusters) submitted to search, the number of spectra/peptides/proteins that were identified (using score thresholds set to maintain a 98% true positive peptide identification rate. The table also notes the difference (as percentages) between the figures obtained for the non-clustered search and the clustered searches of each dataset.

Dataset	Search type	#Spectra searched		# Spectra identified		Peptides identified		Proteins identified	
Human	Non-clust.	0.793 M		86682		21090		6191	
(1 run)	Clustered	0.341 M	-57.0%	99488	+14.8%	21142	+0.2%	6281	+1.5%
Human	Non-clust.	4.0 M		369431		33975		7142	
(5 runs)	Clustered	0.65 M	-83.8%	484913	+31.3%	33175	-2.4%	7143	+0.0%
Human	Non-clust.	11.4 M		815764		59062		8562	
(14 runs)	Clustered	1.85 M	-83.8%	1610667	+97.4%	64512	+9.2%	9104	+6.3%
Shewanella	Non-clust.	14.5 M		1628796		39411		2797	
	Clustered	1.29 M	-91.1%	2889426	+77.4%	43262	+9.8%	2895	+3.5%
Dictyostelium	Non-clust.	1.41 M		272900		40578		6076	
	Clustered	0.71 M	-49.6%	319735	+17.2%	39759	-2.0%	6077	+0.0%
Yeast	Non-clust.	179377		21597		2555		658	
	Clustered	116227	-35.2%	23666	+9.6%	2482	-2.9%	653	-0.8%

runs instead of single run only tripled the number of peptide identifications obtained with the Human data). However, the false identifications are more diverse between runs (since they are spurious database hits generally occurring with lower quality spectra). Thus the growth rate of false identifications accumulated when results of multiple runs are combined is greater than the growth rate of correct peptides. This observation is illustrated well in the results for the human samples in Table 7.5. When searching a single run from the human sample, the F-score threshold needed for 98% true positive peptide identifications was 3.34. With this threshold 86682 of the 793000 spectra (10.9%) were identified in the non-clustered search. When the results of the 14 runs from the human sample were pooled together, in order to achieve a similar rate of 98% the F-score threshold had to be raised to 3.74. With this higher threshold only 815764 of the 11.4 million spectra were identified (7.2%). Had we used the score threshold of 3.34 with the 14 runs, we would have identified 1.32 million spectra, however the peptide accuracy rate would have been only 95.9%.

Table 7.6: Distributions of cluster sizes. Results of clustering 11.4 million spectra from 14 runs of the Human dataset. The left side holds the cluster size distribution for 5.6 million spectra that passed spectral quality filtration and were grouped into 1.85 million clusters. The middle holds the distribution for the subset of clusters that were identified in the database search. The right hand side holds the distribution of "perfect" clustering, in which all the spectra belonging to a single peptide are grouped into a single cluster.

Clust.	All Clusters				Identified Clusters				"Perfect" Clusters			
size	#Clust.	(%)	#Spec.	(%)	#Clust.	(%)	#Spec.	(%)	#Clust.	(%)	#Spec.	(%)
1	1275893	68.9%	1275893	22.7%	143487	53.6%	143487	8.9%	15830	24.6%	15830	1.0%
2	235387	12.7%	470774	8.4%	30993	11.6%	61986	3.8%	8027	12.5%	16054	1.0%
3-5	156917	8.5%	558491	10.0%	31008	11.6%	113935	7.1%	12474	19.4%	47641	3.0%
6-10	58085	3.1%	447501	8.0%	17902	6.7%	138702	8.6%	9447	14.7%	72646	4.5%
11-15	38758	2.1%	506373	9.0%	12484	4.7%	163030	10.1%	4905	7.6%	62629	3.9%
16-25	75335	4.1%	1290671	23.0%	26742	10.0%	461884	28.7%	5169	8.0%	101827	6.4%
26-50	6065	0.32%	210775	3.8%	2636	1.0%	91294	5.7%	4169	6.5%	146599	9.2%
51-100	2590	0.14%	179410	3.2%	1203	0.4%	82743	5.1%	2074	3.2%	143300	9.0%
101-500	1689	0.09%	332853	5.9%	832	0.3%	166587	10.3%	1820	2.8%	375502	23.5%
500+	373	0.02%	335727	6.0%	205	0.1%	187019	11.6%	403	0.6%	617369	38.6%
Total	1851092		5608468		267492		1610667		64318		1599397	

Our clustering experiments support the common view that the existing peptide identification approaches do not identify many spectra in MS/MS datasets. Of the 11.4 million spectra from the human dataset submitted for clustering, 5.6 million passed the spectral quality filtering and ended up being grouped into 1.85 million clusters (see Table 7.6). Only 267492 of these clusters, containing 1.6 million spectra from the original non-clustered dataset were identified in the database search. Thus the majority of the clusters (86.5%) and the majority of the spectra (71.4%) remain unidentified after the database search. Table 7.6 also shows that there is a significant difference between the distribution of cluster sizes in the entire dataset and the distribution of sizes of the identified clusters, with the identified spectra on average belonging to larger clusters. As we mentioned above, our algorithm is not aimed at producing the optimal clustering (i.e., minimal number of clusters). On average, each of the 64318 identified peptides has 4.16 clusters associated with it. It is interesting to note the large range of spectral counts observed for the identified peptides. While most of the peptides have low spectral counts (56.5% of the peptides have 1-5 spectra assigned to them), most of the identified spectra belong to peptides with high redundancy (62.1% of the identified spectra belonged to 2223 peptides, each with at least 100 spectra assigned to it).

7.D Discussion

We presented a practical MS-Clustering algorithm capable of handling large datasets (over ten million spectra) using a single desktop PC. MS-Clustering can lead to a tenfold reduction in the number of spectra submitted to further analysis. With large datasets, searching clusters often yields more peptide and protein identifications than a regular search without clustering (see Table 7.5). These additional identifications can mostly be attributed to the fact that clustering greatly reduces the number of low quality spectra that are submitted to analysis, which in turn reduces the number of spurious database hits to the decoy database. When smaller datasets are clustered (1 million spectra), clustering still gives 2-4 folds reduction in the number of spectra that need to be analyzed, possibly with a small reduction in the number of peptides and proteins identified (typically around 2%), it is not as useful for smaller datasets (below

0.5 million), since this usually leads to some loss of peptide identifications.

Since clustering is usually much faster than a database search, reducing the number of spectra that need to be submitted for analysis leads to a significant reduction in the running time (see Table 7.4). Another benefit of clustering is its ability to single out interesting cases of unidentified spectra that are worthy of further examination. For instance, spectra of peptides with mutations and unexpected PTMs require time-consuming advanced search techniques. Instead of scattering the resources on examination of all unidentified spectra in the dataset (which typically involves the majority of the spectra), we can focus the efforts on the large unidentified clusters which represent the most likely candidates for these interesting peptides. This way we can afford to apply more time-consuming searches to a smaller set of high quality candidates (consensus spectra of large clusters have a high signal-to-noise ratio). In an essence, if searching for these atypical peptides is analogous to searching for a needle in a haystack, clustering can be used to reduce the haystack to an amenable size. Such a reduction can make time-consuming analysis methods like "blind" PTM searches computationally feasible, even for large scale projects with tens of millions of spectra.

With the increasing amount of experimental data being collected and validated, spectrum libraries of identified mass spectra are emerging as a viable method for peptide identification [46, 81, 125, 131, 204, 239]. Spectral libraries contain spectra derived from clusters of spectra from *previously identified* peptides that are compared with the query spectrum to determine a match. The main drawback of spectrum libraries is that they are not applicable to spectra of *previously unidentified* peptides. We propose to extend the notion of spectral libraries by introducing *spectral archives* [1] that contain clusters of unidentified spectra as well. Recently proposed spectral network approach allows one to identify uninterpreted spectra using other uninterpreted spectra (as opposed to using a database) thus opening a possibility to use spectral archives for peptide identifications. Clustering can be viewed as an instrument for constructing spectral archives that can be further interpreted via spectral networks and shotgun protein sequencing [11, 12].

When examining the details of our clustering algorithm we note that it takes

[1] We use the term "archives", since as opposed to libraries that are typically well-annotated collections of books, archives often have many documents that were never looked at, studied, or annotated.

a heuristic approach, and thus might not deliver "optimal" clustering. However, in the mass spectra domain, the payoff for having optimal clustering (as compared to suboptimal) is not high. Often times, clusters get split due to natural variation observed in different instances of spectra of the same peptide. There is no significant advantage to minimizing cluster fragmentation. Whether we have a minimal number of clusters or a slightly larger number, it still represents large savings in time compared to the case when no clustering is performed at all (see Table 7.5). Furthermore, there can be advantages to having several small but more homogenous clusters instead of one larger and more diverse cluster. It is more likely that the peptide in question will get identified at least once when searching several consensus spectra of tighter homogeneous clusters, compared to the case where we have only a single consensus spectrum from a large and noisier cluster.

This chapter, in full, was published as "Clustering Millions of Mass Spectra". A.M. Frank, N. Bandeira, Z. Shen, S. Tanner, S.P. Briggs, R.D. Smith and P.A. Pevzner. *Journal of Proteome Research*. The dissertation author was the primary author of this paper.

8

Interpreting Top-Down Mass Spectra Using Spectral Alignment

8.A Introduction

Most MS-based proteomics research is performed in the "bottom-up" mode in which intact proteins are digested into short peptides and sequenced using tandem mass spectrometry [49, 61, 221]. While suitable for *identification* of the proteins in a sample, this method is deficient when it comes *characterizing* the protein's exact form, such as identifying the locations of post-translational modifications (PTMs), single amino acids polymorphisms, insertions, deletions, etc. [2, 138] In this chapter we examine the alternative "top-down" proteomics approach, in which the whole intact protein is ionized and fragmented [117, 200]. With this approach one gains both information on the whole protein's precursor mass and on the fragments of the entire protein (rather than fragments of individual peptides), which makes it easier to infer the protein's form. For instance, the presence of a PTM can be deduced by observing an increase in the precursor mass and a shift in the masses of the fragment ions containing the PTM.

Top-down proteomics requires high-resolution instrumentation, which is part of the reason why top-down approaches are not used as widely as bottom-up approaches. Consequently, there has been limited top-down data available to researchers and shortage of algorithms for analyzing this data. To date, most of the top-down analysis still requires

significant manual effort.

The only algorithm that is currently available for identifying protein forms from mass spectra of large intact proteins is ProSightPTM [126, 241]. ProSightPTM uses either an intact-mass [220] or a tag filtration based approach [139] to select candidate proteins. It then generates all possible protein forms of the sequence that lie within the specified precursor mass range (e.g., + 2000 Da) and uses a Poisson-model to provide statistically significant matches [145] between the fragment ion mass values in the experimental MS/MS spectrum and the theoretical masses predicted from each candidate protein form. This "absolute mass" search mode is error-tolerant in "Δm" mode [220], which considers the mass difference (Δm) between the molecular weight of the experimental protein and the database candidate, during the search. The candidate expansion method (referred to as "Shotgun Annotation" [164]), allows for careful examination of the putative forms, but leads to an exponential growth in the number of candidate protein forms that need to be considered. This creates the need to curate a custom protein database with a large number of forms even when dealing with a small set of known modifications and locations [164]. This candidate expansion method quickly becomes intractable if one considers more than 20 modifications in a protein form within the context of a highly-annotated eukaryotic database.

In this chapter we demonstrate that the *spectral alignment* algorithm of Pevzner et al. [168, 169] efficiently solves the problem of interpreting top-down spectra and identifying various protein forms. Spectral alignment was recently adapted to solve several bottom-up proteomics computational problems such as a blind database search [223], construction of spectral networks [12], and shotgun protein sequencing [11]. We demonstrate that it can be adapted for top-down proteomics as well. However, while the approach we take is similar to the the one described in refs. [168, 169], there are several issues that need to be addressed to make it applicable top-down mass spectra. In particular, the spectral alignment described in refs. [11, 12, 168, 169, 223] typically deals with 1-2 modifications while top-down spectral alignment may deal with as many as 10-20 modifications to a protein. Another important difference is that with top-down mass spectrometry one often deals with multiple isobaric protein forms in the same spectrum [163]. This biological problem essentially amounts to recovering several alignments

from the same spectrum as opposed to a single spectral alignment described in refs. [168, 169]. In addition one has to address the specifics of top-down mass spectrometry such as the mass measurement errors and weak fragmentation patterns and reflect them in the spectral alignment algorithm.

8.B Materials and Methods

8.B.1 FT-MS/MS Spectra of Histone H4

We analyzed 10 top-down mass spectra acquired on a Q-FT ICR hybrid mass spectrometer from the Kelleher lab. The mass spectra were of different forms of the intact human histone H4 proteins extracted from HeLa cells [165]. Spectra were calibrated externally using an ECD spectrum of bovine ubiquitin. Internal calibration on several selected fragment ions allowed others to be measured within 1-5 ppm of their predicted values. The protein sequence of histone H4 contains 102 amino acids, which add up to 11229.3 Da for the unmodified protein form. The mass spectra were recorded as peak lists of m/z values along with their relative intensities.

8.B.2 The Spectral Alignment Algorithm

The spectral alignment algorithm finds an optimal alignment between a mass spectrum A and a protein sequence B using a specified number of mass shifts (corresponding to post translational modification, mutations, insertions/deletions, etc.). Below we briefly describe the algorithm (see refs. [168, 169] for more details).

We represent a mass spectrum A as a list of n ordered real valued peak masses $a_1 < a_2 < \ldots < a_n$ (we assume that $a_0 = 0$ and a_n represents the protein's precursor mass). The protein sequence B of length m is represented as a list of theoretical peak masses $b_0 < b_1 < b_2 < \ldots < b_m$, where the mass b_i equals the sum of the masses of the first i amino acids in B (we assume $b_0 = 0$ and b_m equals the molecular weight of the unmodified protein form). The mass spectrum A is assumed to be de-convoluted so that each peak is the monoisotopic singly-charged variant of a prefix fragment ion (i.e., we assume that a_i is the mass of the first j amino acids in B for some $1 \leq j \leq m$). We describe below how to create de-convoluted mass spectra from experimental mass

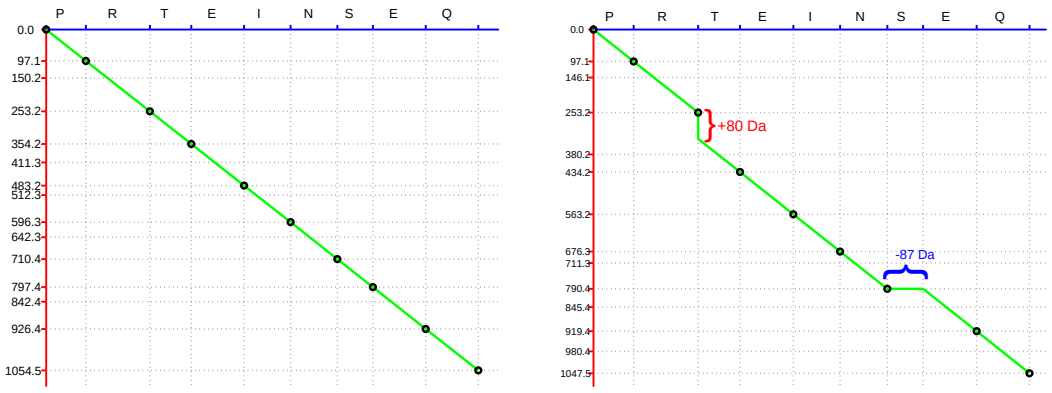


Figure 8.1: Spectral alignment examples. The left displays an alignment of the mass spectrum $A = \{\ldots\}$ with the protein sequence $B = PRTEINSEQ$. On the right we see an alignment of the spectrum A' which contains modifications (+80 Da on T and a deletion of S which comes to -87 Da) with the same protein sequence. The green paths denote the alignments, and the black circles along the paths denote matched pairs of protein prefixes and spectrum peaks.

spectra.

Alignments between A and B can be visualized as paths on a two-dimensional grid as depicted in Figure 8.1. There are three ways in which a path can advance in the grid:

- Diagonally - from a point (a_i, b_j) to a point $(a_{i'}, b_{j'})$, such that $a_{i'} - a_i = b_{j'} - b_j$.
- Vertically - from a point (a_i, b_j) to a point $(a_{i'}, b_j)$, $a_{i'} > a_i$. A vertical step in the alignment represents an addition to the proteins mass, such as a PTM with positive net mass or an insertion.
- Horizontally - from a point (a_i, b_j) to a point $(a_i, b_{j'})$, $b_{j'} > b_j$. A horizontal step in the alignment represents a subtraction to the protein's mass, such as a PTM with a negative net mass or a deletion.

Each path from (a_0, b_0) to (a_n, b_m) is a possible alignment between A and B. However the alignments can differ in quality. The score of an alignment is determined by the shared peak count [168], which equals the number of points of the form (a_i, b_j) that appear on the alignment's path. While the number of possible alignment paths grows exponentially as we increase the number of allowed mass shifts F (each mass shift corresponds to a

vertical or horizontal step in the grid), spectral alignment is an efficient (quadratic) algorithm for finding a maximal scoring alignment (with running time increasing only linearly with F).

We use dynamic programming to find a maximal scoring alignment path in the grid created from the n peaks of mass spectrum A and the set of m prefix masses of protein B. We recursively fill an $n \times m \times F$ array D, in which the value $D_{ij}(f)$ is the highest scoring alignment path from (a_0, b_0) to (a_i, b_j) using at most f mass shifts. After completing the array D, the optimal alignment score is given in $D_{nm}(F)$. In order to obtain a protein form that has that maximal score, we maintain backtracking pointers during the creation of the array D, that denote the predecessor for each cell $D_{ij}(f)$. The pointer in $D_{ij}(f)$ points to the previous cell $D_{i'j'}(f')$ on the optimal path to $D_{i,j}(f)$. See ref. [111] for additional information on using backtracking pointers in dynamic programming.

The array D is created as follows. We say that pairs (a_i, b_i) and $(a_{i'}, b_{i'})$ are *codiagonal* if

$$a_i - a_{i'} = b_j - b_{j'}, \tag{8.1}$$

and we say that $(i, j) < (i', j')$ if $i < i'$ and $j < j'$. We define $diag(i, j)$ as the maximal codaigonal pair of (i, j) such that $diag(i, j) < (i, j)$, i.e., $diag(i, j)$ is the previous point of the form $(a_{i'}, b_{j'})$ on the same diagonal. If no such $(a_{i'}, b_{j'})$ exists, $diag(i, j)$ is set to $(0, 0)$. We define an $n \times m \times F$ array M as

$$M_{ij}(f) = \max_{(i',j') \leq (i,j)} D_{i'j'}(f) \tag{8.2}$$

The recurrence for computing $D_{ij}(f)$ is given by

$$D_{ij}(f) = \max \begin{cases} D_{diag(i,j)}(f) + 1 \\ M_{(i-1,j-1)}(f-1) + 1 \end{cases} \tag{8.3}$$

The recurrence for $M_{ij}(f)$ is given by

$$M_{ij}(f) = \max \begin{cases} D_{ij}(f) \\ M_{i-1,j}(f) \\ M_{i,j-1}(f) \end{cases} \tag{8.4}$$

The starting values of the cells at (0,0) are set to $D_{0,0}(F) = 0$. The number of modifications F affects only one dimension of the dynamic programming array, and increasing F only leads to a linear increase in the number of cells in the array D. Since filling each cell in D requires constant time, the running time of algorithm increases only linearly with the value of F, and not exponentially as with previous approaches. The total running time required for the algorithm to find an optimal alignment is $O(nmF)$, where n is the number of peaks, m is the protein length, and F is the maximal number of mass shifts such as modifications or mutations that occur to the protein form.

8.B.3 Modifying Spectral Alignment for Top-Down Mass Spectra

Deconvolution of Mass Spectra to Prefix Mass Lists

The experimental mass spectra are given as peak lists with m/z values and corresponding intensities. The peak lists contain clusters of isotopic distributions, so prior to running the spectral alignment algorithm, the experimental mass spectra must be deconvoluted to bring them to a form of a list of monoisotopic singly charge prefix masses (i.e., each signal peak is assumed to be the mass of the first j amino acids in protein B for $1 \leq j \leq m$).

To retain the maximal number of signal peaks, we used a manual step in which isotopic peak clusters were extracted from the spectra files. Following that we used a Perl script to convert the isotopic clusters to monoisotopic masses using a least-squared fitting to the theoretical isotopic distributions of fragments of the histone H4 protein which were calculated using the Mercury algorithm [176]. In cases where it was difficult to determine the correct monoisotopic mass due to an inconclusive match to the theoretical isotopic distribution, the script outputted several monoisotopic peaks for the given isotopic cluster. As an alternative to our partially manual method, one could generate the monoisotopic peak lists using existing deconvolution algorithms such as Thrash [100], MassPike [116], or Aid-MS [29].

The monoisotopic peak list contains masses of various types of protein fragments. The prevalent fragments generated with ECD fragmentation are c- and z-ions. Therefore, to create a prefix mass list, we considered each peak with mass m to be both

a c- and a z-fragment, and the corresponding prefix mass was calculated accordingly as $m - 17.026$ for c-ions and $M - m$ for z-ions (where M is the mass of the intact protein).

Accounting for Errors in Mass Measurements in Top-Down Spectra

The spectral alignment algorithm relies on detecting pairs of points (a_i, b_j), $(a_{i'}, b_{j'})$ that are codiagonal (such that $a_{i'} - a_i = b_{j'} - b_j$). Since the protein prefix masses b_j and $b_{j'}$ are calculated directly from the protein sequence, the mass difference $b_{j'} - b_j$ is determined precisely. However, the spectra prefix masses $a_i, a_{i'}$ are calculated from the spectrum's observed peaks, which can have measurement errors. To account for these errors, the peak masses a_i are assigned error margins according to the instrument's accuracy. For example, if the peak masses are measured with 10 ppm accuracy, then a peak with mass 1000 Da has an error tolerance of ± 0.01 Da. Thus, if a peak a_i has an error margin of $\pm\alpha$, and the peak $a_{i'}$ has an error margin of $\pm\alpha'$, then the pairs (a_i, b_j) and $(a_{i'}, b_{j'})$ are assumed to be codiagonal if

$$|a_{i'} - a_i - b_{j'} + b_j| \leq \alpha + \alpha'. \tag{8.5}$$

Eq. 8.5 replaces Eq. 8.1 as the test used by spectral alignment to identify codiagonal pairs of points on the grid.

Accounting for Poor Fragmentation in Top-Down Spectra

The spectral alignment algorithm is designed to find an optimal alignment between a mass spectrum and protein sequence using arbitrary mass shifts. However, in top-down mass spectrometry the fragmentation can be sparse, making it possible for even a single noise peak to lead to unwanted effects such as erroneous peak selection. Figure 8.2 depicts such a case. The correct alignment which uses two +42 Da shifts (corresponding to acetylations on K's) has the same score as an alignment that uses a +25 Da and +59 Da shifts, which do not correspond to plausible modifications. Without proper control, the noise peaks can lead the algorithm to return alignments that have a maximal score but are incorrect from a biological perspective.

To avoid such alignment errors the spectral alignment algorithm can be restricted to use mostly mass shifts that belong to a specific set Δ_{PTMs} of known modifi-

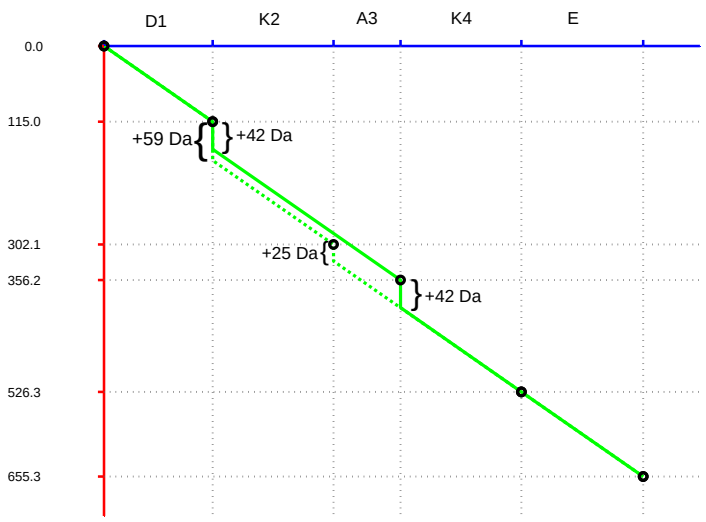


Figure 8.2: Illustration of erroneous peak selection. A single noise peak can lead the algorithm to return maximal scoring alignments that are not biologically plausible. Two equally scoring alignments are shown. The biologically plausible alignment (solid line) has two acetylations on K2 and K4 (42 Da + 42 Da = 84 Da), while the alignment that was selected (dash line) had an equal score but uses two much less plausible modifications, 59 Da on K2 and 25 Da on A3, to reach the same total modification mass of 84 Da.

cations (e.g., oxidations, methylations, etc.), and a small number of general mass shifts that can account for the less common mass shifts (e.g., amino acid substitutions and uncommon PTMs). The choice of which mass shifts to include in Δ_{PTMs} can depend on prior biological knowledge.

In order to account for the different types of mass shifts (general vs. specific), we need to add a dimension to the spectral alignment dynamic programming arrays. Assuming we have at most F_g general modifications and F_s specific modifications from the set Δ_{PTMs}, we construct $n \times m \times F_g \times F_s$ arrays D and M, similar to the ones defined above. Extending the definition of $diag(i,j)$, we define $diag_{(\delta)}(i,j)$ to be the maximal pair of (i',j') such that $(a_i - a_{i'}) - (b_j - b_{j'}) = \delta$. If no such pair (i',j') exists, we set $diag_{(\delta)}(i,j) = (0,0)$. The array cell $D_{ij}(g,s)$ holds the maximal scoring path from (0,0) to position (i,j) using g general shifts and s specific mass shifts from Δ_{PTMs}. The update rule for D (originally described in Eq. 8.3) is extended with an additional term

to handle specific mass shifts in Δ_{PTMs}, and is given by

$$D_{ij}(g,s) = \max \begin{cases} D_{diag(i,j)}(g,s)+1, & \text{// no shift} \\ M_{i-1,j-1}(g-1,s) & \text{// a general shift} \\ D_{diag_{(\delta)}(i,j)}(g,s-1)+1, \quad \delta \in \Delta_{PTMs} & \text{// a specific PTM shift} \end{cases} \tag{8.6}$$

The update rule for M remains similar to Eq. 8.4,

$$M_{ij}(g,s) = \max \begin{cases} D_{ij}(g,s), \\ M_{i-1,j}(g,s), \\ M_{i,j-1}(g,s) \end{cases} \tag{8.7}$$

Note that using these modified update rules increases the running time by a factor that is linear in each of F_s, F_g and $|\Delta_{PTMs}|$. However, the algorithm remains efficient compared to previous attempts which had an exponential growth in running time.

8.B.4 Recovering Multiple Spectral Alignments

As it becomes possible to analyze larger proteins with top-down mass spectrometry, we are more likely to encounter spectra that contain mixtures of several different protein forms [88]. The spectral alignment algorithm can be helpful in detecting these multiple forms. We assume that each protein form has at least one additional peak that differentiates it from the highest scoring form. This leads to the creation of distinct alignment paths that can easily be detected with the spectral alignment algorithm. To facilitate the recovery of additional, possibly suboptimal, protein forms, we need to maintain additional backtracking pointers while filling the dynamic programming tables. Recovering the top t forms, requires that we remember the best t predecessors of each cell $D_{ij}(f)$ when filling the array D.

8.C Results

8.C.1 Identification of Protein Forms Using Spectral Alignment

We implemented the spectral alignment algorithm as a C++ program called MS-TopDown. It is available as open source from `http://proteomics.bioprojects.org`.

Table 8.1: Results for spectral alignment on 10 histone H4 ECD spectra. Following the convention in ref [165], the modifications are labeled as "a" (acetylation) or "m" (methylation) along with the lysine on which they occur. If the modification is preceded by 2 or 3 that denotes a double or triple modification, for example 2mK20 denotes two methyls (or a dimethyl) on the Lysine at position 20. (*) All proteins contain an additional N-terminal acetylation (42 Da) and all forms except for spectrum #10 have a +32 Da modification towards the C-terminal (a double oxidation of M84). (**) Some of the spectra contain evidence of more than one protein form. For each spectrum, the form listed in the table is one (of possibly few) that matched the maximal number of peaks.

Spec.	mod mass(*)	# Peaks in raw spectrum	# Peaks deconvoluted	# Peaks matched	Matched form(**)
1	+186	172	161	33	aK12 + aK16 + 2mK20
2	+186	164	126	30	2 from {aK8,aK12,aK16} + 2mK20
3	+172	297	125	27	aK12 + aK16 + mK20
4	+186	167	157	35	aK5 + aK12 + 2mK20
5	+172	166	155	16	(aK8 or aK12) + aK16 + mK20
6	+202	490	159	27	aK5 + aK12 +2mK20 +16Da (G48-V87)
7	+158	75	81	13	(aK5 or aK8 or aK12) + aK16
8	+172	447	49	13	2 from {aK8,aK12,aK16} + mK20
9	+186	416	111	24	aK8 + aK12 + 2mK20
10	+168	233	101	25	aK5 + aK12 + aK15

The algorithm outputs optimal spectral alignment for varying numbers of specific and general mass shifts (below we explain how to determine the correct number of mass shifts for each spectrum). It can also return annotated peak lists for the identified protein forms. The set of specific modifications contained 20 common PTMs including acetylation (42 Da), methylation (14 Da), oxidation (16 Da), double oxidation (32 Da), etc. The time required to run the algorithm on each spectrum is less than half of a second on a desktop PC. For the ease of presentation, the masses we discuss below were either given as nominal integer values or are rounded to one place past the decimal point. However, the algorithm uses high precision mass measurements, and thus if given data obtained with sufficiently high-resolution, the algorithm could distinguish between modifications with very similar masses such as acetylations and tri-methylations.

The spectral alignment results are summarized in Table 8.1. For each of the 10 test spectra the table includes the number of peaks used in the original spectrum, the number of prefix peaks retained after the spectrum was deconvoluted, the number of peak matches contained in the optimal alignment, and a protein form that corresponds to the maximal alignment. Note that all discovered protein forms had an acetylated

$N-$terminal and all forms except for spectrum #10 had a +32 Da modification towards the C-terminal (a double oxidation of M84), which are not mentioned in the table. For each spectrum the table lists a single modified protein form, however many of the spectra contain additional forms which differ in the location of the modifications. With each spectrum, the form predicted by the spectral alignment algorithm is the same as one of the forms predicted through manual analysis of the data [165]. Not all modifications sites were directly flanked with peaks (which could help pinpoint the location of the modification to a specific amino acid), however when coming to assign locations to the modification we relied on the fact that the acetylations and methylations of the histone H4 occur primarily on the lysine residues. Thus if spectral alignment revealed that there is a +42 Da mass shift between G9 and A15, we were able to assign it to K12, since that is the only lysine in that range. However, in cases were there was more than one lysine residue in the range, we could not determine which was the residue with the modification (in which case the table contains entries like "aK8 or aK12").

8.C.2 Reliability of spectral alignment

We compared our results on the test spectra with simulations in which we ran the algorithm on 1000 random proteins (by shuffling the amino acids in the histone H4 sequence). When all the mass shifts were allowed to be arbitrary, the average number of matched peaks in such a case was 10.4 with $\sigma = 1.2$. With such values, some of the low signal-to-noise spectra, such as the ones with 13 matched peaks in the table, would be difficult to distinguish from a random match. However, when we restricted the search by enforcing the solutions contain at most 2 arbitrary mass shifts (while the rest had to conform to the list of 20 common PTMs), the random matched scores had a distribution with mean 8.4 and $\sigma = 1.04$. Restricting the solution to contain mass shifts that only correspond to the 20 known PTMs reduced the average random match score to 4.6 with $\sigma = 1.3$. Thus, the parameters of the algorithm can be set to confidently identify protein forms even with spectra that have a low signal-to-noise ratio. Note that the chances of having a high scoring match to a random protein in the database can be further reduced by using tag filtration [139, 147], such as the method used by ProSightPTM [126].

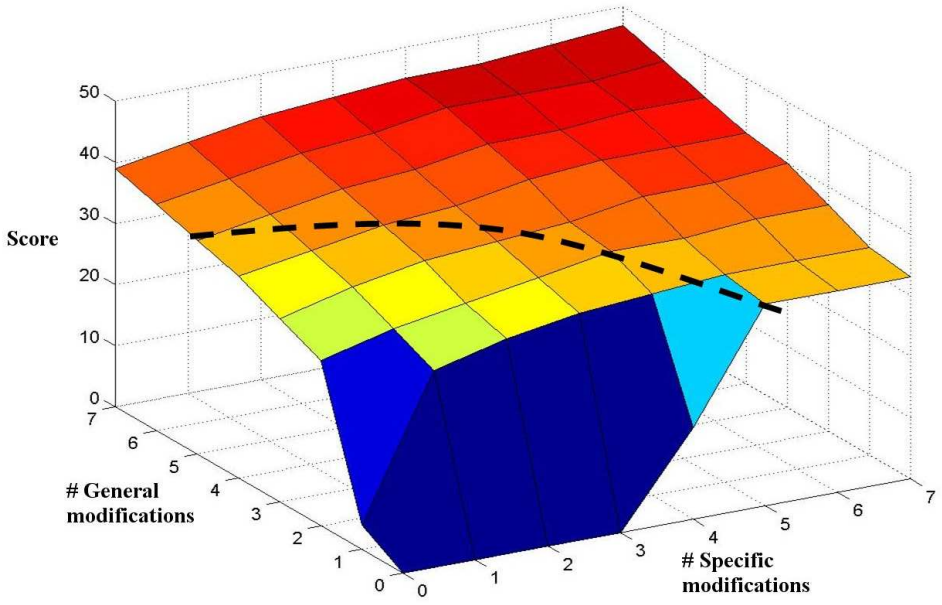


Figure 8.3: The scores obtained when aligning spectrum #1 using various numbers of general and specific modifications. A score of 0 denotes that no alignment could be found. The dotted line separates the regions of fast and slow score growth and designates the correct number of modifications (5). The alignment's modifications consist of an N-terminal acetylation, acetylations on K12 and K16, two methyls on K20 and double oxidation on M84.

8.C.3 Finding the correct number of modifications

In most cases, prior to running spectral alignment we do not know the total number of mass shifts F and how many of these belong to the specified set of modifications (F_s) and how many can be general shifts (F_g). However, in the process of building the dynamic programming arrays for the spectral alignment, we compute alignment scores for paths using various combinations of F_g and F_s. By examining the scores obtained using different numbers of shifts, we can select the parameters that represent the true number of modifications. As a rule of thumb, real mass shifts should lead to a significant contribution to the score. If the increase to the score is not significant, this is most likely an indication that the additional mass shift does not represent a real

modification that is part of the protein form (this is especially true for general shifts, since if they only increase the peak count by 1, it is likely to be a case of an opportunistic inclusion of a noise peak into the alignment).

Figure 8.3 depicts a 3-D plot of the alignment scores obtained using various combinations of specific and general mass shifts when performing spectral alignment on spectrum #1. When using only specific shifts, we need at least 4 shifts to make an alignment with score 15, though using 5 shifts (which is the correct number of modifications) gives a higher score of 33. Increasing the number of specific shifts beyond 5 did not increase the alignment score. The score difference between paths with 4 and 5 specific modifications is significant (the alignment includes 33-15=18 new peaks), so we assume that the solution with 5 modifications is a better match. Since with this protein form all mass shifts belong to the set of specific shifts Δ_{PTMs}, using general shifts did not lead to significantly better alignments (for instance using 3 specific modifications and 2 general modifications only gave a score of 34). Adding general modifications beyond the 5 specific PTMs did not improve the score significantly either. This fact is reflected in the figure by the dotted line (representing 5 modifications) that separates between the region of fast score growth (too few modifications) and the region of slow score growth (too many modifications). All points along the line represent the same optimal solution, with the general modifications getting assigned masses that correspond to the specific modifications that occur to the protein form (methylations and acetylations).

8.C.4 Identifying multiple protein forms in a single mass spectrum

Figure 8.4 gives an example of a spectrum of a mixture of two isobaric protein forms that we were able to recover using the spectral alignment algorithm. Each of the two forms matched 25 peaks (trying to recover a third form gave only 12 matched peaks). Both forms have acetylations on the $N-$terminal and on $K12$ and $K16$, but differ on the location of the third acetylation. The first form (depicted in the figure as the green path) has an acetylation on K5, while the other form (corresponding to the red path), has an acetylation on K8. The mass spectrum has ample evidence to support the presence of these two forms: There are 3 peaks that are only matched in the green path (at masses 569.3, 626.3 and 683.3) which correspond to a +84 Da shift at K5, G6

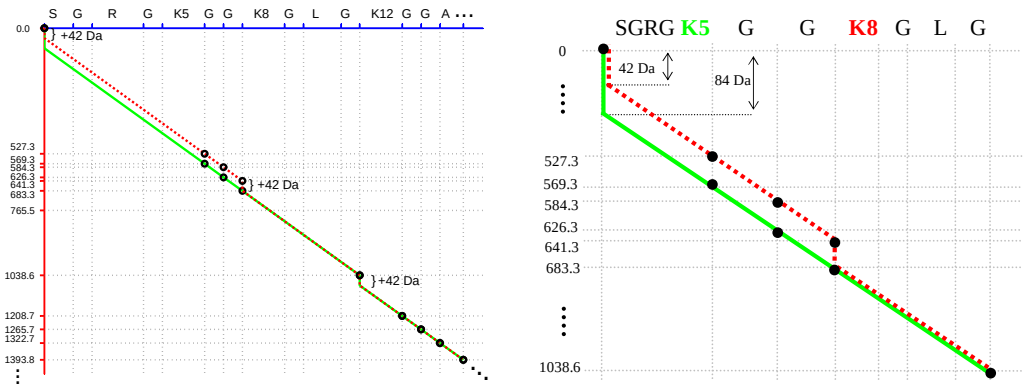


Figure 8.4: Spectral alignment with two protein forms. The left figure displays the spectral alignment grid for spectrum #10 (the grid is drawn to scale and displays the first 15 amino acids). There are 3 modifications displayed, an N-terminal acetylation, and two additional possible acetylations. The first occurs either at K5 on K8, and the second occurs at K12. Accordingly, two paths are shown, the green path corresponds to the protein form that has an acetylation on K5 and the dashed red path corresponds to the protein form with an acetylation on K8. The figure on the right illustrates the portion of the grid in which the two paths differ (it is not drawn to scale).

and G7, respectively. Similarly, there are 3 peaks that are only matched in the red path (at masses 527.3, 584.3 and 641.3) which correspond to a +42 Da shift at K5, G6, and G7, respectively. After K8 both forms have the same number of acetylations, and they display the same modification pattern for the reminder of the protein sequence. So from that point on (mass 1038.6), the two paths are merged.

Though in many cases, spectral alignment is capable of recovering several forms from a given spectrum, these are not always genuine protein forms. There are a couple rules-of-thumb that can be used to distinguish between the false and genuine protein form alignments. First, the alignment score of the additional forms should be close to the score of best alignment. In the example mentioned above (Figure 8.4) the two highest scoring paths have a score of 25, while the third path has a significantly lower score of only 12, and should thus not be considered a genuine protein form - or at least we should assume that the spectrum lacks sufficient evidence to support the third form. A second rule that can help eliminate many false protein forms is to insist that there should be a significant difference between the new form and the previously recovered ones. This difference should manifest itself in at least one new peak that points to a modification

that distinguishes the new form from the previous ones. To reduce the probability that incorporating a noise peak is what led to the difference between forms, we need to make sure that the additional peak supports a known modification type, and not an arbitrary mass shift. Since the conditions for including additional forms are not cut-and-dried, it is recommended to use manual validation of the data with borderline cases.

8.D Discussion

The spectral alignment algorithm is a fast and accurate method for finding modified protein forms. The algorithm can discover alignments with arbitrary mass shifts, which makes it suitable for detecting a wide class of modifications: known and novel PTMs, single amino acid polymorphisms, insertions/deletions, etc. The alignments can be found efficiently since the algorithm's running time increases linearly in the number of modifications. The algorithm also has polynomial running time, which makes it suitable both for interpreting large proteins and for searching a protein sequence database in cases of samples from unknown proteins.

We demonstrated how spectral alignment can be used to determine the locations of modifications on proteins using top-down mass spectra. In the design of our algorithm we addressed several issues that are particular to top-down mass spectra such as mass inaccuracies, large sets of possible PTMs and the presence of multiple protein forms (modification patterns) in the same spectrum. Our algorithm produces the same results as the time consuming manual analysis. The capability to rapidly and accurately determine locations of modifications becomes more and more important as the data volume grows and the length of the proteins and the number of modifications increases [200]. With our data, most of analysis effort was focused on the first 30 amino acids of the histone H4, however there are cases of spectra containing peaks for hundreds of amino acids [88], which make manual analysis difficult.

The work we present here is a preliminary exploration into the use of spectral alignment for protein form determination. There are many ways in which this work can be extended and improved. Currently we use a simple shared peak count to score alignments, however more complex scoring schemes can be devised that also take into

account the intensity of the peaks and the presence of multiple supporting fragment ions such as different charge states, neutral losses and internal fragments. Another related challenge would be to devise a model for assigning p-values to the spectrum alignment results using these new scores, which also take into account the number mass shifts that were used in the alignment. This will become especially important when analyzing proteins for which we do not have prior knowledge about the type of modifications that we are likely to encounter. In such cases we would have to use general unrestricted mass shifts which could lead to spurious alignments.

Another interesting avenue to explore would be to combine results of both bottom-up and top-down sequencing of the same analyzed protein to determine a more exact characterization of protein forms [21, 205]. In the context of spectral alignment, bottom-up peptide identifications can help fill in the gaps of incomplete fragmentation observed in the top-down spectra. This can be done by forcing the spectral alignment algorithm to comply with the restrictions on PTM locations (or lack of) that are observed in the peptide identifications. The algorithm will then find a protein form that has the maximal number of matched peaks in the MS spectrum of the intact protein and is completely consistent with the peptides identified from the protein digestion.

This chapter, in full, was published as "Interpreting Top-Down Mass Spectra Using Spectral Alignment". A.M. Frank, J.J. Pesavento, C.A. Mizzen, N.L. Kelleher, and P.A. Pevzner. *Analytical Chemistry*, 80:2499-2505, 2008. The dissertation author was the primary author of this paper.

References

[1] Adomavicius, G. and A. Tuzhilin (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering 17*, 734–749.

[2] Aebersold, R. and M. Mann (2003). Mass spectrometry-based proteomics. *Nature 422*, 198–207.

[3] Aho, A. and M. Corasick (1975). Efficient string matching: an aid to bibliographic search. *Communications of the ACM 18*, 333–340.

[4] Altschul, S., W. Gish, W. Miller, E. Myers, and D. Lipman (1990, Oct). Basic local alignment search tool. *J. Mol. Biol. 215*(3), 403–410.

[5] Ansong, C., S. Purvine, J. Adkins, M. Lipton, and R. Smith (2008). Proteogenomics: needs and roles to be filled by proteomics in genome annotation. *Brief Funct Genomic Proteomic 7*, 50–62.

[6] Auerbach, D., S. Thaminy, M. Hottiger, and I. Stagljar (2002). The post-genomic era of interactive proteomics: facts and perspectives. *Proteomics 2*, 611–23.

[7] Baczek, T., P. Wiczling, M. Marszall, Y. Heyden, and R. Kaliszan (2005). Prediction of peptide retention at different HPLC conditions from multiple linear regression models. *Journal of Proteome Research 4*, 555–563.

[8] Bafna, V. and N. Edwards (2001). SCOPE: a probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinformatics 17 Suppl 1*, 13–21.

[9] Bafna, V. and N. Edwards (2003). On de-novo interpretation of tandem mass spectra for peptide identification. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology*, pp. 9–18.

[10] Balgley, B., T. Laudeman, L. Yang, T. Song, and C. Lee (2007). Comparative Evaluation of Tandem MS Search Algorithms Using a Target-Decoy Search Strategy. *Mol Cell Proteomics 6*, 1599–1608.

[11] Bandeira, N., K. Clauser, and P. Pevzner (2007). Shotgun protein sequencing: Assembly of ms/ms spectra from mixtures of modified proteins. *Mol. Cell. Proteom. 6*, 1123–1134.

[12] Bandeira, N., D. Tsur, A. Frank, and P. Pevzner (2007). Protein identification by spectral networks analysis. *PNAS 104*, 6140–6145.

[13] Bartels, C. (1990). Fast algorithm for peptide sequencing by mass spectroscopy. *Biomedical and Environmental Mass Spectrometry 19*(6), 363–8.

[14] Barton, S., S. Richardson, D. Perkins, I. Bellahn, T. Bryant, and J. Whittaker (2007). Using statistical models to identify factors that have a role in defining the abundance of ions produced by tandem ms. *Analytical Chemistry 79*, 5601–5607.

[15] Beer, I., E. Barnea, T. Ziv, and A. Admon (2004). Improving large-scale proteomics by clustering of mass spectrometry data. *Proteomics 4*, 950–60.

[16] Bern, M., Y. Cai, and D. Goldberg (2007). Lookup peaks: A hybrid of de novo sequencing and database search for protein identification by tandem mass spectrometry. *Analytical Chemistry 79*, 1393–1400.

[17] Bern, M. and D. Goldberg (2005). EigenMS: De novo analysis of peptide tandem mass spectra by spectral graph partitioning. In *Proceedings of the Ninth annual International Conference on Research in Computational Molecular Bilogy (RECOMB 2005)*, pp. 357–372.

[18] Bern, M. and D. Goldberg (2006). De novo analysis of peptide tandem mass spectra by spectral graph partitioning. *Journal of Computational Biology 13*, 36478.

[19] Bern, M., D. Goldberg, W. McDonald, and J. Yates, III (2004). Automatic Quality Assessment of Peptide Tandem Mass Spectra. *Bioinformatics 20*, i49–54.

[20] Biemann, K. (1988). Contributions of mass spectrometry to peptide and protein structure. *Biomed Environ Mass Spectrom 16*, 99111.

[21] Borchers, C., R. Thapar, E. Petrotchenko, M. Torres, J. Speir, M. Easterling, Z. Dominski, and W. F. Marzluff (2006). Combined top-down and bottom-up proteomics identifies a phosphorylation site in stem-loop-binding proteins that contributes to high-affinity RNA binding. *PNAS 103*, 3094–3099.

[22] Breci, L., E. Hattrup, M. Keeler, J. Letarte, R. Johnson, and P. Haynes (2005). Comprehensive proteomics in yeast using chromatographic fractionation, gas phase fractionation, protein gel electrophoresis, and isoelectric focusing. *Proteomics 5*, 2018–2028.

[23] Breci, L., D. Tabb, J. Yates, III, and V. Wysocki (2003). Cleavage n-terminal to proline: Analysis of a database of peptide tandem mass spectra. *Anal. Chem. 75*, 1963–1971.

[24] Budnik, B. A., M. L. Nielsen, J. V. Olsen, K. Haselmann, P. Hörth, H. W., and R. A. Zubarev (2002). Can relative cleavage frequencies in peptides provide additional sequence information? *Int. J. Mass Spectrom 219*, 283–294.

[25] Cannon, W., K. Jarman, B.-J. Webb-Robertson, D. Baxter, C. Oehmen, K. Jarman, A. Heredia-Langner, K. Auberry, and G. Anderson (2005). Comparison of probability and likelihood models for peptide identification from tandem mass spectrometry data. *J. of Proteome Res. 4*, 1687–1698.

[26] Cargile, B., J. Sevinsky, A. Essader, J. Stephenson, J.L., and J. Bundy (2005). Immobilized pH Gradient Isoelectric Focusing as a First-Dimension Separation in Shotgun Proteomics. *J Biomol Tech 16*, 181–189.

[27] Castellana, N., S. Payne, Z. Shen, M. Stanke, S. Briggs, and V. Bafna (2008). Validation and expansion of the arabidopsis gene annotation. Submitted.

[28] Chait, B. (2006). Mass Spectrometry: Bottom-Up or Top-Down? *Science 314*, 65–66.

[29] Chen, L., S. Sze, and H. Yang (2006). Automated intensity descent algorithm for interpretation of complex high-resolution mass spectra. *Anal. Chem. 78*, 5006–5018.

[30] Chen, T., M. Kao, M. Tepel, J. Rush, and G. Church (2001). A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *J. Comput. Biol. 8*, 325–337.

[31] Cheng, C. and M. Gross (2000). Applications and mechanisms of charge-remote fragmentation. *Mass Spectrometry Reviews 19*, 398–420.

[32] Choudhary, J., W. Blackstock, D. Creasy, and J. Cottrell (2001). Matching peptide mass spectra to EST and genomic DNA databases. *Trends Biotechnol 19*, S17–S22.

[33] Clauser, K., P. Baker, and A. Burlingame (1999). Role of accurate mass measurement (+/- 10 ppm) in protein identification strategies employing ms or ms/ms and database searching. *Anal. Chem. 71*, 2871–2882.

[34] Cole, R. (2000). Some tenets pertaining to electrospray ionization mass spectrometry. *Journal of Mass Spectrometry 35*, 763–772.

[35] Colinge, J. (2007). Peptide fragment intensity statistical modeling. *Analytical Chemistry 79*, 7286–7290.

[36] Colinge, J., I. Cusin, S. Reffas, E. Mahe, A. Niknejad, P.-A. Rey, H. Mattou, M. Moniatte, and L. Bougueleret (2005). Experiments in searching small proteins in unannotated large eukaryotic genomes. *Journal of Proteome Research 4*, 167–174.

[37] Colinge, J., J. Magnin, T. Dessingy, M. Giron, and A. Masselot (2003). Improved peptide charge state assignment. *Proteomics 3*, 1434–1440.

[38] Colinge, J., A. Masselot, I. Cusin, E. Mahé, A. Niknejad, G. Argoud-Puy, S. Reffas, N. Bederr, A. Gleizes, P. Rey, and L. Bougueleret (2004). High-performance peptide identification by tandem mass spectrometry allows reliable automatic data processing in proteomics. *Proteomics 4*, 1977–1984.

[39] Colinge, J., A. Masselot, M. Giron, T. Dessingy, and J. Magnin (2003). OLAV: Towards high-throughput tandem mass spectrometry data identification. *Proteomics 3*, 1454–1463.

[40] Collins, M. (2000). Discriminative reranking for natural language parsing. In *Proc. 17th International Conf. on Machine Learning*, pp. 175–182. Morgan Kaufmann, San Francisco, CA.

[41] Collins, M. and N. Duffy (2002). New ranking algorithms for parsing and tagging: Kernels over discrete structures. In *In 40th Annual Meeting of the ACL, Philadelphia, July 2002.*, pp. 263–270.

[42] Collins, M. and T. Koo (2005). Discriminative reranking for natural language parsing. *Computational Linguistics 31*(1), 25–70.

[43] Craig, R. and R. Beavis (2003). A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid Communications in Mass Spectrometry 17*, 2310–2316.

[44] Craig, R. and R. Beavis (2004). TANDEM: matching proteins with tandem mass spectra. *Bioinformatics 20*, 1466–1467.

[45] Craig, R., J. Cortens, and R. Beavis (2005). The use of proteotypic peptide libraries for protein identification. *Rapid Commun Mass Spectrom 19*, 1844–1850.

[46] Craig, R., J. Cortens, D. Fenyo, and R. Beavis (2006). Using annotated peptide mass spectrum libraries for protein identification. *J. of Proteome Research 5*, 1843 –1849.

[47] Creasy, D. and J. Cottrell (2002). Error tolerant searching of uninterpreted tandem mass spectrometry data. *Proteomics 2*, 1426–1434.

[48] Csonka, I., B. Paizs, G. Lendvay, and S. Suhai (2000). Proton mobility in protonated peptides: a joint molecular orbital and rrkm study. *Rapid Communications in Mass Spectrometry 14*, 417–431.

[49] Dancík, V., T. Addona, K. Clauser, J. Vath, and P. Pevzner (1999). De novo peptide sequencing via tandem mass spectrometry. *J. Comput. Biol. 6*, 327–342.

[50] Day, R., A.Borziak, and A. Gorin (2004). PPM-Chain de novo peptide identification program comparable in performance to sequest. In *Proceedings of 2004 IEEE Computational Systems in Bioinformatics (CSB 2004)*, pp. 505–508.

[51] Delahunty, C. and J. Yates III (2005). Protein identification using 2D-LC-MS/MS. *Methods 35*, 248–55.

[52] Desiere, F., E. Deutsch, A. Nesvizhskii, P. Mallick, N. King, J. Eng, A. Aderem, R. Boyle, S. Brunner, E. Donohoe, N. Fausto, E. Hafen, L. Hood, M. Katze, K. Kennedy, F. Kregenow, H. Lee, B. Lin, D. Martin, J. Ranish, D. Rawlings, L. Samelson, Y. Shiio, J. Watts, B. Wollscheid, M. Wright, W. Yan, L. Yang, E. Yi, H. Zhang, and R. Aebersold (2005). Integration with the human genome of peptide sequences obtained by high-throughput mass spectrometry. *Genome Biol 6*, R9.

[53] Dongré, A., J. Jones, A. Somogyi, and V. Wysocki (1996). Influence of Peptide Composition, Gas-Phase Basicity and Chemical Modification on Fragmentation Efficiency: Evidence for the Mobile Proton Model. *J. Am. Chem. Soc. 118*, 8365–8374.

[54] Duda, R. and P. Hart (1973). *Pattern Classification and Scene Analysis*. Wiley-Interscience.

[55] Dutta, D. and T. Chen (2007). Speeding up tandem mass spectrometry database search: Metric embeddings and fast near neighbor search. *Bioinformatics 23*, 612–618.

[56] Eddes, J., E. Kapp, D. Frecklington, L. Connolly, M. Layton, R. Moritz, and R. Simpson (2002). CHOMPER: A bioinformatic tool for rapid validation of tandem mass spectrometry search results associated with high-throughput proteomic strategies. *PROTEOMICS 2*, 1097–1103.

[57] Edman, P. and G. Begg (1967). A protein sequenator. *Eur. J. Biochem 1*, 80–91.

[58] ElAribi, H., G. Orlova, A. Hopkinson, and K. Siu (2004). Gas-phase fragmentation reactions of protonated aromatic amino acids: Concomitant and consecutive neutral eliminations and radical cation formations. *Journal of Physical Chemistry A 108*, 3844–3853.

[59] Elias, J., F. Gibbons, O. King, F. Roth, and S. Gygi (2004). Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nat. biotech. 22*, 214–219.

[60] Elias, J. and S. Gygi (2007). Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods 4*, 207–214.

[61] Eng, J., A. McCormack, and J. Yates, III (1994). An Approach to Correlate Tandem Mass-Spectral Data of Peptides with Amino Acid Sequences in a Protein Database. *Journal Of The American Society For Mass Spectrometry 5*, 976–989.

[62] Everitt, S., S. Landau, and M. Leese (2001). *Cluster Analysis* (4th ed.). Arnold.

[63] Fenn, J., M. Mann, C. Meng, S. Wong, and C. Whitehouse (1989). Electrospray Ionization for Mass Spectrometry of Large Biomolecules. *Science 246*, 64–71.

[64] Fenyo, D. and R. Beavis (2003). A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Anal. Chem. 75*, 768–74.

[65] Fenyo, D., B. Phinney, and R. Beavis (2007). Determining the overall merit of protein identification data sets: rho-diagrams and rho-scores. *J Proteome Res. 6*, 1997–2004.

[66] Fermin, D., B. Allen, T. Blackwell, R. Menon, M. Adamski, Y. Xu, P. Ulintz, G. Omenn, and D. States (2006). Novel gene and gene model detection using a whole genome open reading frame analysis in proteomics. *Genome Biol 7*, R35.

[67] Fernández-de Cossío, J., J. Gonzalez, and V. Besada (1995, Aug). A computer program to aid the sequencing of peptides in collision-activated decomposition experiments. *Comput. Appl. Biosci. 11*, 427–434.

[68] Field, H., D. Fenyö, and R. Beavis (2002). RADARS, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in a relational database. *PROTEOMICS 2*, 36–47.

[69] Fischer, B., V. Roth, F. Roos, J. Grossmann, S. Baginsky, P. Widmayer, W. Gruissem, and J. Buhmann (2005). NovoHMM: A Hidden Markov Model for de novo peptide sequencing. *Anal. Chem. 77*, 7265–7273.

[70] Flikka, K., L. Martens, J. Vandekerckhove, and I. Gevaert, K. amd Eidhammer (2006). Improving the reliability and throughput of mass spectrometry-based proteomics by spectrum quality filtering. *Proteomics 6*, 2086–2094.

[71] Frank, A., N. Bandeira, Z. Shen, S. Tanner, S. Briggs, R. Smith, and P. Pevzner (2008). Clustering millions of tandem mass spectra. *J. of Proteome Research 7*, 113–122.

[72] Frank, A., J. Pesavento, C. Mizzen, N. Kelleher, and P. Pevzner (2008). Interpreting top-down mass spectra using spectral alignment. *Anal. Chem. 80*, 2499–2505.

[73] Frank, A. and P. Pevzner (2005). Pepnovo: De novo peptide sequencing via probabilistic network modeling. *Anal. Chem. 77*, 964–973.

[74] Frank, A., M. Savitski, M. Nielsen, R. Zubarev, and P. Pevzner (2007). De novo peptide sequencing and identification with precision mass spectrometry. *J. of Proteome Research 6*, 114–123.

[75] Frank, A., S. Tanner, V. Bafna, and P. Pevzner (2005). Peptide sequence tags for fast database search in mass-spectrometry. *J. of Proteome Research 4*, 1287–95.

[76] Frank, A., S. Tanner, and P. Pevzner (2005). Peptide sequence tags for fast database search in mass-spectrometry. In *Proceedings of the Ninth annual International Conference on Research in Computational Molecular Bilogy (RECOMB 2005)*, pp. 326–341.

[77] Freund, Y., R. Iyer, R. Schapire, and Y. Singer (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research 4*, 933–969.

[78] Freund, Y. and L. Mason (1999). The alternating decision tree algorithm. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 124–133.

[79] Freund, Y. and R. Schapire (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pp. 23–37.

[80] Freund, Y. and R. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences 55*, 119–139.

[81] Frewen, F., G. Merrihew, C. Wu, W. Stafford Noble, and M. MacCoss (2006). Analysis of peptide ms/ms spectra from large-scale proteomics experiments using spectrum libraries. *Anal. Chem. 78*, 5678 – 5684.

[82] Friedman, J., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics 2*, 337374.

[83] Gabelica, V. and E. De Pauw (2005). Internal energy and fragmentation of ions produced in electrospray sources. *Mass Spectrometry Reviews 24*, 566–587.

[84] Geer, L., S. Markey, J. Kowalak, L. Wagner, M. Xu, D. Maynard, X. Yang, W. Shi, and S. Bryant (2004). Open mass spectrometry search algorithm. *J. Proteome Res. 3*, 958–64.

[85] Gevaert, K. and J. Vandekerckhove (2000). Protein identification methods in proteomics. *Electrophoresis 21*, 1145–1154.

[86] Gu, C., G. Tsaprailis, L. Breci, and V. Wysocki (2000). Selective gas-phase cleavage at the peptide bond c-terminal to aspartic acid in fixed-charge derivatives of asp-containing peptides. *Analytical Chemistry 72*, 5804–5813.

[87] Gupta, N., S. Tanner, N. Jaitly, J. Adkins, M. Lipton, R. Edwards, M. Romine, A. Osterman, V. Bafna, R. Smith, and P. Pevzner (2007). Whole proteome analysis of post-translational modifications: applications of mass-spectrometry for proteogenomic annotation. *Genome Res. 17*, 1362–1377.

[88] Han, X., M. Jin, K. Breuker, and F. McLafferty (2006). Extending top-down mass spectrometry to proteins with masses greater than 200 kilodaltons. *Science 314*, 109–12.

[89] Han, Y., B. Ma, and K. Zhang (2005). SPIDER: software for protein identification from sequence tags with de novo sequencing error. *J Bioinform. Comput. Biol. 3*, 697–716.

[90] Harrison, A. (2003). Fragmentation reactions of protonated peptides containing glutamine or glutamic acid. *J Mass Spectrom 38*, 174–87.

[91] Harrison, A. and T. Yalcin (1997). Proton mobility in protonated amino acids and peptides. *International Journal of Mass Spectrometry and Ion Processes 165*, 339–347.

[92] Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer.

[93] Havilio, M., Y. Haddad, and Z. Smilansky (2003). Intensity-based statistical scorer for tandem mass spectrometry. *Anal. Chem. 75*, 435–444.

[94] Havilio, M. and A. Wool (2007). Large-scale unrestricted identification of post-translation modifications using tandem mass spectrometry. *Analytical Chemistry 79*, 1362–1368.

[95] Henzel, W., T. Billeci, J. Stults, S. Wong, C. Grimley, and C. Watanabe (1993). Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases. *Proc. Natl. Acad. Sci. USA 90*, 5011–5.

[96] Hernandez, P., R. Gras, J. Frey, and R. Appel (2003). Popitam: towards new heuristic strategies to improve protein identification from tandem mass spectrometry data. *Proteomics 3*, 870–8.

[97] Higdon, R., J. Hogan, G. V. Belle, and E. Kolker (2005). Randomized sequence databases for tandem mass spectrometry peptide and protein identification. *OMICS 9*, 364–379.

[98] Higgs, R., M. Knierman, A. BonnerFreeman, L. Gelbert, S. Patil, and J. Hale (2007). Estimating the statistical significance of peptide identifications from shotgun proteomics experiments. *Journal of Proteome Research 6*, 1758–1767.

[99] Horn, D., R. Zubarev, and F. McLafferty (2000a). Automated de novo sequencing of proteins by tandem high-resolution mass spectrometry. *PNAS 97*, 10313 – 10317.

[100] Horn, D., R. Zubarev, and F. McLafferty (2000b). Automated reduction and interpretation of high resolution electrospray mass spectra of large molecules. *J. Am. Soc. Mass Spectrom 11*, 320–332.

[101] Huang, Y., J. Triscari, G. Tseng, L. Pasa-Tolic, M. Lipton, R. Smith, and V. Wysocki (2005). Statistical characterization of the charge state and residue dependence of low-energy CID peptide dissociation patterns. *Anal. Chem. 77*, 5800–5813.

[102] Huang, Y., J. Triscari, V. Wysocki, L. Pasa-Tolic, G. Anderson, M. Lipton, and R. Smith (2004). Dissociation behaviors of doubly-charged tryptic peptides: Correlation of gas-phase cleavage abundance with ramachandran plots. *J. Am. Chem. Soc. 126*, 3034–3035.

[103] Huang, Y., V. Wysocki, D. Tabb, and J. Yates, III (2002). The influence of histidine on cleavage c-terminal to acidic residues in doubly protonated tryptic peptides. *Int. J. Mass Spectrom. 219*, 233–244.

[104] Ishikawa, K. and Y. Niva (1986). Computer-aided peptide sequencing by fast atom bombardment mass spectrometry. *Biomed. Environ. Mass Spectrom. 13*, 373–380.

[105] Jaffe, J., H. Berg, and G. Church (2004). Proteogenomic mapping as a complementary method to perform genome annotation. *Proteomics 4*, 59–77.

[106] Jain, A., M. Murty, and P. Flynn (1999). Data clustering: a review. *ACM Computing Surveys 31*, 264–323.

[107] Jensen, F. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag.

[108] Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of Knowledge Discovery in Databases*.

[109] Johnson, R. and K. Biemann (1989). Computer program (seqpep) to aid in the interpretation of high-energy collision tandem mass spectra of peptides. *Biomed. Environ. Mass Spectrom 18*, 945–957.

[110] Jones, J., A. Dongre, A. Somogyi, and V. Wysocki (1994). Sequence dependence of peptide fragmentation efficiency curves determined by electrospray ionization/surface-induced dissociation mass spectrometry. *Journal of the American Chemical Society 116*, 8368–8369.

[111] Jones, N. and P. Pevzner (2004). *An introduction to Bioinformatics algorithms*. MIT Press.

[112] Käll, L., J. Storey, M. MacCoss, and W. Noble (2008). Posterior error probabilities and false discovery rates: Two sides of the same coin. *Journal of Proteome Research 7*, 40–44.

[113] Kalume, D., S. Peri, R. Reddy, J. Zhong, M. Okulate, N. Kumar, and A. Pandey (2005). Genome annotation of anopheles gambiae using mass spectrometry-derived data. *BMC Genomics 6*, 128.

[114] Kapp, A., F. Schtz, L. Connolly, J. Chakel, J. Meza, C. Miller, D. Fenyo, J. Eng, J. Adkins, G. Omenn, and R. Simpson (2005). An evaluation, comparison, and accurate benchmarking of several publicly available ms/ms search algorithms: Sensitivity and specificity analysis. *Proteomics 5*, 3475–3490.

[115] Kapp, E., F. Schutz, G. Reid, J. Eddes, R. Moritz, R. O'Hair, T. Speed, and R. Simpson (2003). Mining a tandem mass spectrometry database to determine the trends and global factors influencing peptide fragmentation. *Anal. Chem. 75*, 6251–6264.

[116] Kaur, P. and P. OConnor (2006). Algorithms for automatic interpretation of high resolution mass spectra. *J. Am. Soc. Mass Spectrom. 17*, 459–468.

[117] Kelleher, N. L. (2004). Top down proteomics. *Anal. Chem. 76*, 197A–203A.

[118] Keller, A., A. Nesvizhskii, E. Kolker, and R. Aebersold (2002). Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem. 74*, 5383–5392.

[119] Keller, A., S. Purvine, A. Nesvizhskii, S. Stolyar, D. Goodlett, and E. Kolker (2002). Experimental protein mixture for validating tandem mass spectral analysis. *OMICS 6*, 207–212.

[120] Kim, S., N. Gupta, N. Bandeira, and P. Pevzner (2008). Spectral dictionaries: Integrating de novo peptide sequencing with database search of tandem mass spectra. submitted.

[121] Kim, S., N. Gupta, and P. Pevzner (2008). The generating function of tandem mass spectra: a new approach to evaluating statistical significance of peptide identifications. submitted.

[122] Klammer, A., C. Wu, M. MacCoss, and W. Noble (2005). Peptide charge state determination for low-resolution tandem mass spectra. In *Proc IEEE Comput. Syst. Bioinform. Conf.*, pp. 175–185.

[123] Krijgsveld, J., S. Gauci, W. Dormeyer, and A. Heck (2006). In-gel isoelectric focusing of peptides as a tool for improved protein identification. *Journal of Proteome Research 5*, 1721–1730.

[124] Krokhin, O., S. Ying, J. Cortens, D. Ghosh, V. Spicer, W. Ens, K. Standing, R. Beavis, and J. Wilkins (2006). Use of peptide retention time prediction for protein identification by off-line reversed-phase hplc-maldi ms/ms. *Analytical Chemistry 78*, 6265–6269.

[125] Lam, H., E. Deutsch, J. Eddes, J. Eng, S. King, N. Stein, and R. Aebersold (2007). Development and validation of a spectral library searching method for peptide identification from ms/ms. *Proteomics 7*, 655–667.

[126] LeDuc, R., G. Taylor, Y. Kim, T. Januszyk, L. Bynum, J. Sola, J. Garavelli, and N. Kelleher (2004). ProSight PTM: an integrated environment for protein identification and characterization by top-down mass spectrometry. *Nucl. Acids Res. 32*, W340–345.

[127] Leipzig, J., P. Pevzner, and S. Heber (2004). The Alternative Splicing Gallery (ASG): bridging the gap between genome and transcriptome. *Nucleic Acids Res. 32*, 3977–83.

[128] Lioe, H. and R. O'Hair (2005). Neighbouring group processes in the deamination of protonated phenylalanine derivatives. *Org. Biomol. Chem. 3*, 3618 – 3628.

[129] Lioe, H., R. O'Hair, and G. Reid (2004). Gas-phase reactions of protonated tryptophan. *J Am Soc Mass Spectrom 15*, 65–76.

[130] Liu, C., B. Yan, Y. Song, Y. Xu, and L. Cai (2006). Peptide sequence tag-based blind identification of post-translational modifications with point process model. *Bioinformatics 22*, e307–313.

[131] Liu, J., A. Bell, J. Bergeron, C. Yanofsky, B. Carrillo, C. Beaudrie, and R. Kearney (2007). Methods for peptide identification by spectral comparison. *Proteome Sci. 5*, 3.

[132] Lu, B. and T. Chen (2003a). A suboptimal algorithm for de novo peptide sequencing via tandem mass spectrometry. *J Comput. Biol. 10*, 1–12.

[133] Lu, B. and T. Chen (2003b). A suffix tree approach to the interpretation of tandem mass spectra: applications to peptides of non-specific digestion and post-translational modifications. *Bioinformatics 19*(suppl_2), ii113–121.

[134] Lubeck, O., C. Sewell, S. Gu, X. Chen, and D. Cai (2002). New computational approaches for de novo peptide sequencing from MS/MS experiments. *IEEE Proc. on Challenges in Biomedical Informatics 90*, 1868–1874.

[135] M., K. and F. Hillenkamp (1988). Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons. *Anal. Chem. 60*, 2299–2301.

[136] Ma, B., K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby, and G. Lajoie (2003). PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Commun. Mass Spectrom. 17*, 2337–2342.

[137] MacCoss, M., C. Wu, and J. Yates (2002). Probability-based validation of protein identifications using a modified sequest algorithm. *Analytical Chemistry 74*, 5593–5599.

[138] Mallick, P., M. Schirle, S. Chen, M. Flory, H. Lee, D. Martin, J. Ranish, B. Raught, R. Schmitt, T. Werner, K. B., and R. Aebersold (2007). Computational prediction of proteotypic peptides for quantitative proteomics. *Nature Biotech. 25*, 125–131.

[139] Mann, M. and M. Wilm (1994). Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Anal. Chem. 66*, 4390–4399.

[140] Marshall, A. and C. Hendrickson (2002). Fourier transform ion cyclotron resonance detection: Principles and experimental configurations. *Int. J. Mass Spectrom. 215*, 59–75.

[141] Marshall, A. and F. Verdun (1990). *Fourier Transforms In NMR, Optical, And Mass Spectrometry : A User's Handbook*. Elsevier.

[142] Martin, D., J. Eng, A. Nesvizhskii, A. Gemmill, and R. Aebersold (2005). Investigation of neutral loss during collision-induced dissociation of peptide ions. *Analytical Chemistry 77*, 4870–4882.

[143] Masselon, C., L. Pasa-Tolic, N. Tolic, G. Anderson, B. Bogdanov, A. Vilkov, Y. Shen, R. Zhao, W. Qian, M. Lipton, D. Camp, and R. Smith (2005). Targeted comparative proteomics by liquid chromatography-tandem fourier ion cyclotron resonance mass spectrometry. *Anal. Chem. 77*, 400–406.

[144] Matthiesen, R., M. Trelle, P. Hojrup, J. Bunkenborg, and O. Jensen (2005). VEMS 3.0: Algorithms and computational tools for tandem mass spectrometry based identification of post-translational modifications in proteins. *Journal of Proteome Research 4*, 2338–2347.

[145] Meng, F., L. Cargile, B.J. nad Miller, J. Johnson, and N. Kelleher (2001). Informatics and Multiplexing of Intact Protein Identification in Bacteria and The Archaea. *Nat. Biotechnol. 19*, 952–956.

[146] Mo, L., D. Dutta, Y. Wan, and T. Chen (2007). MSNovo: A Dynamic Programming Algorithm for de Novo Peptide Sequencing via Tandem Mass Spectrometry. *Analytical Chemistry 79*, 4870–4878.

[147] Mørtz, E., P. O'Connor, P. Roepstorff, N. Kelleher, T. Wood, and M. M. F.W. McLafferty (1996). Sequence tag identification of intact proteins by matching tandem mass spectral data against sequence data bases. *PNAS 93*, 8264 – 8267.

[148] Narasimhan, C., D. Tabb, N. VerBerkmoes, M. Thompson, R. Hettich, and E. Uberbacher (2005). MASPIC: Intensity-based tandem mass spectrometry scoring scheme that improves peptide identification at high confidence. *Analytical Chemistry 77*, 7581–7593.

[149] Nesvizhskii, A., A. Keller, E. Kolker, and R. Aebersold (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem. 75*, 4646–4658.

[150] Nesvizhskii, A., F. Roos, J. Grossmann, M. Vogelzang, J. Eddes, W. Gruissem, S. Baginsky, and R. Aebersold (2006). Dynamic spectrum quality assessment and iterative computational analysis of shotgun proteomic data: toward more efficient identification of post-translational modifications, sequence polymorphisms, and novel peptides. *Mol Cell Proteomics 5*, 652–670.

[151] Nesvizhskii, A., O. Vitek, and R. Aebersold (2007). Analysis and validation of proteomic data generated by tandem mass spectrometry. *Nature Methods 4*, 787 – 797.

[152] Ng, J. and P. Pevzner (2008). Algorithm for identification of fusion proteins via mass spectrometry. *Journal of Proteome Research 7*, 89–95.

[153] O'Hair, R. (2000). The role of nucleophile-electrophile interactions in the unimolecular and bimolecular gas-phase ion chemistry of peptides and related systems. *Journal of Mass Spectrometry 35*, 1377–1381.

[154] Olsen, J., L. de Godoy, G. Li, B. Macek, P. Mortensen, R. Pesch, A. Makarov, O. Lange, S. Horning, and M. Mann (2005). Parts per million mass accuracy on an orbitrap mass spectrometer via lock mass injection into a C-trap. *Mol. Cell. Proteomics 4*, 2010–2021.

[155] Olson, M., J. Epstein, and A. Yergey (2006). De novo peptide sequencing using exhaustive enumeration of peptide composition. *J. Am. Soc. Mass Spectrom 17*, 1041–1049.

[156] Owens, K. (1992). Application of correlation analysis techniques to mass spectral data. *Applied Spectroscopy Reviews 27*, 1–49.

[157] Paizs, B. and S. Suhai (2005). Fragmentation pathways of protonated peptides. *Mass Spectrometry Reviews 24*, 508–548.

[158] Pandey, A. and M. Mann (2000). Proteomics to study genes and genomes. *Nature 405*, 837–846.

[159] Papayannopoulos, I. (1995). The interpretation of collision-induced dissociation tandem mass spectra of peptides. *Mass Spectrometry Reviews 14*, 49–73.

[160] Payne, S., M. Yau, M. Smolka, S. Tanner, H. Zhou, and V. Bafna (2008). Phosphorylation specific ms/ms scoring for rapid and accurate phospho-proteome analysis. submitted.

[161] Pazzani, M. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review 13*, 393–408.

[162] Perkins, D., D. Pappin, D. Creasy, and J. Cottrell (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis 20*, 3551–3567.

[163] Pesavento, J., B. Garcia, J. Streeky, N. Kelleher, and C. Mizzen (2007). Mild Performic Acid Oxidation Enhances Chromatographic and Top Down Mass Spectrometric Analyses of Histones. *Mol. Cell. Proteomics 6*, 1510–1526.

[164] Pesavento, J., Y. Kim, G. Taylor, and N. Kelleher (2004). Shotgun annotation of histone modifications: A new approach for streamlined characterization of proteins by top down mass spectrometry. *J. of Am Chem. Soc. 126*, 3386–3387.

[165] Pesavento, J., C. Mizzen, and N. Kelleher (2006). Quantitative analysis of modified proteins and their positional isomers by tandem mass spectrometry: Human histone h4. *Anal. Chem. 78*, 4271–4280.

[166] Petritis, K., L. Kangas, P. Ferguson, G. Anderson, L. Pasa-Tolic, M. Lipton, K. Auberry, E. Strittmatter, Y. Shen, R. Zhao, and R. Smith (2003). Use of artificial neural networks for the accurate prediction of peptide liquid chromatography elution times in proteome analyses. *Analytical Chemistry 75*, 1039–1048.

[167] Pevtsov, S., I. Fedulova, H. Mirzaei, C. Buck, and X. Zhang (2006). Performance evaluation of existing de novo sequencing algorithms. *J. Prot. Research 5*, 3018–3028.

[168] Pevzner, P., V. Dancík, and C. Tang (2000). Mutation-tolerant protein identification by mass spectrometry. *J. Comput. Biol. 7*, 777–787.

[169] Pevzner, P., Z. Mulyukov, V. Dancík, and C. Tang (2001). Efficiency of Database Search for Identification of Mutated and Modified Proteins via Mass Spectrometry. *Genome Res. 11*, 290–299.

[170] Polce, M., D. Ren, and C. Wesdemiotis (2000). Dissociation of the peptide bond in protonated peptides. *Journal of Mass Spectrometry 35*, 1391–1398.

[171] Prince, J., M. Carlson, R. Wang, P. Lu, and E. Marcotte (2004). *Nat. Biotech. 22*, 471–472.

[172] Purvine, S., N. Kolker, and E. Kolker (2004). Spectral quality assessment for high-throughput tandem mass spectrometry proteomics. *OMICS 8*, 255–265.

[173] Purvine, S., A. Picone, and E. Kolker (2004). Standard mixtures for proteome studies. *OMICS 8*, 79–92.

[174] Ramakrishnan, S., R. Mao, A. Nakorchevskiy, J. Prince, W. Willard, W. Xu, E. Marcotte, and D. Miranker (2006). A fast coarse filtering method for peptide identification by mass spectrometry. *Bioinformatics 22*, 1524–1531.

[175] Razumovskaya, J., V. Olman, D. Xu, E. Uberbacher, N. VerBerkmoes, R. Hettich, and Y. Xu (2004). A computational method for assessing peptide-identification reliability in tandem mass spectrometry analysis with sequest. *PROTEOMICS 4*, 961–969.

[176] Rockwood, A. and S. Van Orden (1996). Ultrahigh-speed calculation of isotope distributions. *Anal. Chem. 68*, 2027–2030.

[177] Roepstorff, P. and J. Fohlman (1984). Proposal for a common nomenclature for sequence ions in mass spectra of peptides. *Biomed. Mass Spectrom. 11*, 601.

[178] Romine, M., D. Elias, M. Monroe, K. Auberry, R. Fang, J. Fredrickson, G. Anderson, R. Smith, and M. Lipton (2004). Validation of Shewanella oneidensis MR-1 small proteins by AMT tag-based proteome analysis. *OMICS 8*, 239–54.

[179] Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review 65*, 386–408.

[180] Sadygov, R., H. Liu, and J. Yates (2004). Statistical models for protein validation using tandem mass spectral data and protein amino acid sequence databases. *Analytical Chemistry 76*, 1664–1671.

[181] Sadygov, R., J. Wohlschlegel, S. Park, T. Xu, and J. Yates, III (2006). Central limit theorem as an approximation for intensity-based scoring function. *Anal. Chem. 78*, 89–95.

[182] Sadygov, R. and J. Yates, III (2003). A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. *Anal. Chem. 75*, 3792–3798.

[183] Sakurai, T., T. Matsuo, H. Matsuda, , and I. Katakuse (1984). Paas 3: A computer program to determine probable sequence of peptides from mass spectrometric data. *Biomed. Mass Spectrom 11*, 396–399.

[184] Savitski, M., F. Kjeldsen, M. Nielsen, and R. Zubarev (2007). Relative specificities of water and ammonia losses from backbone fragments in collision-activated dissociation. *Journal of Proteome Research 6*, 2669–2673.

[185] Savitski, M., M. Nielsen, F. Kjeldsen, and R. Zubarev (2005). Proteomics-grade de novo sequencing approach. *J. Proteome Res. 4*, 2348–2354.

[186] Savitski, M., M. Nielsen, and R. Zubarev (2005). New data base-independent, sequence tag-based scoring of peptide ms/ms data validates mowse scores, recovers below threshold data, singles out modified peptides, and assesses the quality of ms/ms techniques. *Mol. Cell. Proteomics. 4*, 1180–8.

[187] Savitski, M., M. Nielsen, and R. Zubarev (2006). ModifiComb, a New Proteomic Tool for Mapping Substoichiometric Post-translational Modifications, Finding Novel Types of Modifications, and Fingerprinting Complex Protein Mixtures. *Mol Cell Proteomics 5*, 935–948.

[188] Schapire, R. and Y. Singer (1999). Improved boosting using confidence-rated predictions. *Machine Learning 37*, 297–336.

[189] Schlosser, A. and W. Lehmann (2000). Five-membered ring formation in unimolecular reactions of peptides: a key structural element controlling low-energy collision-induced dissociation of peptides. *Journal of Mass Spectrometry 35*, 1382–1390.

[190] Schölkopf, B. and A. Smola (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

[191] Schutz, F., E. Kapp, R. Simpson, and T. Speed (2003). Deriving statistical models for predicting peptide tandem ms product ion intensities. *Biochem. Soc. Trans. 31*, 1479–1483.

[192] Searle, B., S. Dasari, M. Turner, A. Reddy, D. Choi, P. Wilmarth, A. McCormack, L. David, and S. Nagalla (2004). High-throughput identification of proteins and unanticipated sequence modifications using a mass-based alignment algorithm for ms/ms de novo sequencing results. *Anal. Chem. 76*, 2220–2230.

[193] Sevinsky, J., B. Cargile, M. Bunger, F. Meng, N. Yates, R. Hendrickson, and J. Stephenson, Jr. (2008). Whole genome searching with shotgun proteomic data: Applications for genome annotation. *Journal of Proteome Research 7*, 80–88.

[194] Shevchenko, A., A. Loboda, S. Sunyaev, A. Shevchenko, P. Bork, W. Ens, and K. Standing. (2001). Charting the proteomes of organisms with unsequenced genomes by MALDI-Quadrupole Time-of Flight Mass Spectrometry and BLAST homology searching. *Anal. Chem. 73*, 1917–1926.

[195] Shevchenko, A., S. Sunyaev, A. Liska, P. Bork, and A. Shevchenko (2003). Nanoelectrospray tandem mass spectrometry and sequence similarity searching for identification of proteins from organisms with unknown genomes. *Methods Mol. Biol. 211*, 221–234.

[196] Shewchuk, J. (1994). An introduction to the conjugate gradient method without the agonizing pain. `http://www-2.cs.cmu.edu/~jrs/jrspapers.html`.

[197] Shilov, I., S. Seymour, A. Patel, A. Loboda, W. Tang, S. Keating, C. Hunter, L. Nuwaysir, and D. Schaeffer (2007). The Paragon Algorithm, a Next Generation Search Engine That Uses Sequence Temperature Values and Feature Probabilities to Identify Peptides from Tandem Mass Spectra. *Mol. Cell. Proteomics 6*, 1638–1655.

[198] Siegel, M. and N. Bauman (1988). An efficient algorithm for sequencing peptides using fast atom bombardment mass spectral data. *Biomed. Environ. Mass Spectrom. 15*, 333–343.

[199] Siepel, A., M. Diekhans, B. Brejova, L. Langton, M. Stevens, C. Comstock, C. Davis, B. Ewing, S. Oommen, C. Lau, H. Yu, J. Li, B. Roe, P. Green, D. Gerhard, G. Temple, D. Haussler, and M. Brent (2007). Targeted discovery of novel human exons by comparative genomics. *Genome Res. 17*, 1763–73.

[200] Siuti, N. and N. Kelleher (2007). Decoding protein modifications using top-down mass spectrometry. *Nature Methods 4*, 817–821.

[201] Siuzdak, G. (2003). *The expanding role of mass spectrometry in biotechnology.* MCC Press.

[202] Spengler, B. (2004). De novo sequencing, peptide composition analysis, and composition-based sequencing: a new strategy employing accurate mass determination by fourier transform ion cyclotron resonance mass spectrometry. *J. Am. Soc. Mass Spectrom 15*, 703–14.

[203] Stark, A., M. Lin, P. Kheradpour, J. Pedersen, L. Parts, J. Carlson, M. Crosby, M. Rasmussen, S. Roy, A. Deoras, G. Ruby, E. Brennecke, H. Curators, Berkeley, E. Hodges, A. Hinrichs, A. Caspi, B. Paten, S. Park, M. Han, M. Maeder, B. Polansky, B. Robson, S. Aerts, J. van Helden, B. Hassan, D. Gilbert, D. Eastman, M. Rice, M. Weir, M. Hahn, Y. Park, C. Dewey, L. Pachter, J. Kent, D. Haussler, E. Lai, D. Bartel, G. Hannon, T. Kaufman, M. E, A. Clark, D. Smith, S. Celniker, W. Gelbart, and M. Kellis (2007). Discovery of functional elements in 12 drosophila genomes using evolutionary signatures. *Nature 450*, 219–232.

[204] Stein, S. and D. Scott (1994). Optimization and testing of mass spectral library search algorithms for compound identification. *J. Am. Soc. Mass. Spectrom. 5*, 859–866.

[205] Strader, M., N. VerBerkmoes, D. Tabb, H. Connelly, J. Barton, B. Bruce, D. Pelletier, B. Davison, R. Hettich, F. Larimer, and G. Hurst (2004). Characterization of the 70s ribosome from rhodopseudomonas palustris using an integrated "top-down" and "bottom-up" mass spectrometric approach. *J. Proteome Res. 3*, 965–978.

[206] Summerfield, S. and S. Gaskell (1997). Fragmentation efficiencies of peptide ions following low energy collisional activation. *International Journal of Mass Spectrometry and Ion Processes 165*, 509–521.

[207] Sun, S., K. Meyer-Arendt, B. Eichelberger, R. Brown, C.-Y. Yen, W. Old, K. Pierce, K. Cios, N. Ahn, and K. Resing (2007). Improved Validation of Peptide MS/MS Assignments Using Spectral Intensity Prediction. *Mol. Cell. Proteomics 6*, 1–17.

[208] Sunyaev, S., A. Liska, A. Golod, A. Shevchenko, and A. Shevchenko (2003). MultiTag: multiple error-tolerant sequence tag search for the sequence-similarity identification of proteins by mass spectrometry. *Anal. Chem. 75*, 1307–1315.

[209] Syka, J., J. Coon, M. Schroeder, J. Shabanowitz, and D. Hunt (2004). Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry. *Proceedings of the National Academy of Sciences 101*, 9528–9533.

[210] Tabb, D., Y. Huang, V. Wysocki, and J. Yates, III (2004). Influence of basic residue content on fragment ion peak intensities in low-energy collision-induced dissociation spectra of peptides. *Anal. Chem. 76*, 1243–1248.

[211] Tabb, D., M. MacCoss, C. Wu, S. Anderson, and J. Yates, III (2003). Similarity among tandem mass spectra from proteomic experiments: detection, significance, and utility. *Anal. Chem. 75*, 2470–2477.

[212] Tabb, D., A. Saraf, and J. Yates, III (2003). GutenTag: High-throughput sequence tagging via an empirically derived fragmentation model. *Anal. Chem. 75*, 6415–6421.

[213] Tabb, D., L. Smith, L. Breci, V. Wysocki, D. Lin, and J. Yates, III (2003). Statistical characterization of ion trap tandem mass spectra from doubly charged tryptic peptides. *Anal. Chem. 75*, 1155–1163.

[214] Tabb, D., M. Thompson, G. Khalsa-Moyers, N. VerBerkmoes, and W. McDonald (2005). MS2Grouper: Group assessment and synthetic replacement of duplicate proteomic tandem mass spectra. *J. Am. Soc. Mass Spec. 16*, 1250–1261.

[215] Tang, H., R. Arnold, P. Alves, Z. Xun, D. Clemmer, M. Novotny, J. Reilly, and P. Radivojac (2006). A computational approach toward label-free protein quantification using predicted peptide detectability. *Bioinformatics 22*, e481–488.

[216] Tang, X., P. Thibault, and R. Boyd (1993). Fragmentation reactions of multiply-protonated peptides and implications for sequencing by tandem mass spectrometry with low-energy collision-induced dissociation. *Analytical Chemistry 65*(20), 2824–2834.

[217] Tanner, S., S. H. Payne, S. Dasari, Z. Shen, P. A. Wilmarth, L. L. David, W. F. Loomis, S. P. Briggs, and V. Bafna (2008). Accurate annotation of peptide modifications through unrestrictive database search. *Journal of Proteome Research 7*, 170–181.

[218] Tanner, S., Z. Shen, J. Ng, L. Florea, R. Guig, S. Briggs, and V. Bafna (2007). Improving gene annotation using peptide mass spectrometry. *Genome Res. 17*, 231–239.

[219] Tanner, S., H. Shu, A. Frank, M. Mumby, P. Pevzner, and V. Bafna (2005). Inspect: Fast and accurate identification of post-translationally modified peptides from tandem mass spectra. *Anal. Chem. 77*, 4626–4639.

[220] Taylor, G., Y. Kim, A. Forbes, F. Meng, R. McCarthy, and N. Kelleher (2003). Web and database software for identification of intact proteins using "top down" mass spectrometry. *Anal. Chem. 75*, 4081–4086.

[221] Taylor, J. and R. Johnson (1997). Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. *Rapid Commun. Mass Spectrom 11*, 1067–1075.

[222] Taylor, J. and R. Johnson (2001). Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Anal. Chem. 73*, 2594–2604.

[223] Tsur, D., S. Tanner, E. Zandi, V. Bafna, and P. Pevzner (2005). Identification of post-translational modifications via blind search of mass-spectra. *Nature Biotechnology 23*, 1562–2567.

[224] Ulintz, P., J. Zhu, Z. Qin, and P. Andrews (2006). Improved Classification of Mass Spectrometry Database Search Results Using Newer Machine Learning Approaches. *Mol Cell Proteomics 5*, 497–509.

[225] Vaisar, T. and J. Urban (1996). Probing the proline effects in CID of protonated peptides. *J. Mass Spectrom. 31*, 11851187.

[226] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer.

[227] Venable, J. and J. Yates (2004). Impact of ion trap tandem mass spectra variability on the identification of peptides. *Anal Chem 76*, 2928–2937.

[228] Wan, X., I. Vidavsky, and M. Gross (2002). Comparing similar spectra: from similarity index to spectral contrast angle. *J. Am. Soc. Mass. Spectrom. 13*, 85–88.

[229] Wan, Y. and T. Chen (2006). PepHMM: A hidden Markov model based scoring function for tandem mass spectrometry. *Anal. Chem. 78*, 432–7.

[230] Waridel, P., A. Frank, V. Thomas, H. Surendranath, S. Sunyaev, P. Pevzner, and A. Shevchenko (2007). Sequence similarity-driven proteomics in organisms with unknown genomes by lc-ms/ms and automated de novo sequencing. *Proteomics 7*, 2318–29.

[231] Washburn, M., D. Wolters, and J. Yates, III (2001). Large scale analysis of the yeast proteome via multidimensional protein identification technology. *Nature Biotechnology 19*, 242–247.

[232] Wells, J. and S. McLuckey (2005). Collision-induced dissociation (CID) of peptides and proteins. *Meth. Enzymol. 402*, 148–85.

[233] Wielsch, N., H. Thomas, V. Surendranath, P. Waridel, A. Frank, P. Pevzner, and A. Shevchenko (2006). Rapid validation of protein identifications with the borderline statistical confidence via de novo sequencing and ms blast searches. *J. Proteome Res. 5*, 2448 –2456.

[234] Wysocki, V., G. Tsaprailis, L. Smith, and L. Breci (2000). Mobile and localized protons: A framework for understanding peptide dissociation. *J. Mass Spectrom. 35*, 1399–1406.

[235] Yague, J., A. Paradela, M. Ramos, S. Ogueta, A. Marina, F. Barahona, J. Lopez de Castro, and J. Vazquez (2003). Peptide rearrangement during quadrupole ion trap fragmentation: Added complexity to ms/ms spectra. *Analytical Chemistry 75*, 1524–1535.

[236] Yates, III, J., J. Eng, and A. McCormack (1995). Mining genomes: correlating tandem mass spectra of modified and unmodified peptides to sequences in nucleotide databases. *Anal. Chem. 67*, 3202–3210.

[237] Yates, III, J., J. Eng, A. McCormack, and D. Schieltz (1995). Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. *Anal Chem 67*, 1426–1436.

[238] Yates, III, J., P. Griffin, L. Hood, and J. Zhou (1991). Computer aided interpretation of low energy ms/ms mass spectra of peptides. In *Techniques in Protein Chemistry II, Villafranca, J.J. ed*, pp. 477–485.

[239] Yates, III, J., S. Morgan, P. Gatlin, C.L. amd Griffin, and J. Eng (1998). Method to compare collision-induced dissociation spectra of peptides: Potential for library searching and subtractive analysis. *Anal. Chem. 70*, 3557–3565.

[240] Yoshida, T. (1998). Calculation of peptide retention coefficients in normal-phase liquid chromatography. *Journal of Chromatography A 808*, 105–112.

[241] Zamdborg, L., R. LeDuc, K. Glowacz, Y. Kim, V. Viswanathan, I. Spaulding, B. Early, E. Bluhm, B. Shannee, and N. Kelleher (2007). ProSight PTM 2.0: improved protein identification and characterization for top down mass spectrometry. *Nucleic Acids Research 35*, W701W706.

[242] Zhang, N., R. Aebersold, and B. Schwikowski (2002). ProbID: A probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics 2*, 1406–1412.

[243] Zhang, Z. (2004). Prediction of low-energy collision-induced dissociation spectra of peptides. *Anal. Chem. 76*(14), 3908–3922.

[244] Zhang, Z. (2005). Prediction of low-energy collision-induced dissociation spectra of peptides with three or more charges. *Anal. Chem. 77*, 6364–6373.

[245] Zhang, Z. and J. McElvain (2000). De novo peptide sequencing by two-dimensional fragment correlation mass spectrometry. *Anal. Chem. 72*, 2337–2350.

[246] Zidarov, D., P. Thibault, M. Evans, , and M. Bertrand (1990). Determination of the primary structure of peptides using fast atom bombardment mass spectrometry. *Biomed. Environ. Mass Spectrom. 19*, 13–16.

[247] Zubarev, R., N. Kelleher, and F. McLafferty (1998). Electron capture dissociation of multiply charged protein cations. a nonergodic process. *Journal of the American Chemical Society 120*, 3265–3266.